Machine Learning, 58, 39–77, 2005
2005 Springer Science + Business Media, Inc. Manufactured in The Netherlands.

ROC 'n' Rule Learning—Towards a Better Understanding of Covering Algorithms

JOHANNES FÜRNKRANZ fuernkranz@informatik.tu-darmstadt.de
TU Darmstadt, Knowledge Engineering Group, Hochschulstraße 10, D-64289 Darmstadt, Germany

PETER A. FLACH peter.flach@bristol.ac.uk
Department of Computer Science, University of Bristol, Woodland Road, Bristol BS8 1UB, UK

Editor: Tom Fawcett

Abstract. This paper provides an analysis of the behavior of separate-and-conquer or covering rule learning algorithms by visualizing their evaluation metrics and their dynamics in coverage space, a variant of ROC space. Our results show that most commonly used metrics, including accuracy, weighted relative accuracy, entropy, and Gini index, are equivalent to one of two fundamental prototypes: precision, which tries to optimize the area under the ROC curve for unknown costs, and a cost-weighted difference between covered positive and negative examples, which tries to find the optimal point under known or assumed costs. We also show that a straightforward generalization of the *m*-estimate trades off these two prototypes. Furthermore, our results show that stopping and filtering criteria like CN2's significance test focus on identifying significant deviations from random classification, which does not necessarily avoid overfitting. We also identify a problem with Foil's MDL-based encoding length restriction, which proves to be largely equivalent to a variable threshold on the recall of the rule. In general, we interpret these results as evidence that, contrary to common conception, pre-pruning heuristics are not very well understood and deserve more investigation.

Keywords: inductive rule learning, ROC analysis, rule evaluation metrics, coverage space

1. Introduction

Most rule learning algorithms for classification problems follow the so-called *separate-and-conquer* or *covering* strategy, i.e., they learn one rule at a time, each of them explaining (*covering*) a part of the training examples. The examples covered by the last learned rule are removed from the training set (*separated*) before subsequent rules are learned (before the remaining training examples are *conquered*). Typically, these algorithms operate in a *concept learning* framework, i.e., they expect positive and negative examples for an unknown concept. From these training data, they learn a set of rules that describe the underlying concept, i.e., that explain all (or most) of the positive examples and (almost) none of the negative examples.[1]

Various approaches that adhere to this framework differ in the way single rules are learned. Common to all algorithms is that they have to use a metric or heuristic function for evaluating the quality of a candidate rule. A survey of the rule learning literature yields a vast number of rule learning metrics that have been proposed for this purpose. Many of them have justifications based in statistics, information theory, or related fields, but their relation to each other is not very well understood.

In this paper, we attempt to shed some light upon this issue by investigating commonly used evaluation metrics. Our basic tool for analysis will be visualization in coverage spaces. Coverage space is quite similar to ROC space, the two-dimensional plane in which the operating characteristics of classifiers are visualized. Our analysis will show that most commonly used heuristics, including accuracy, weighted relative accuracy, entropy, and Gini index, are equivalent to one of two fundamental prototypes: precision, which tries to optimize the area under the ROC curve for unknown costs, and a cost-weighted difference between covered positive and negative examples, which tries to find the optimal point under known or assumed costs. We also show that a straightforward generalization of the m-estimate trades off these two prototypes.

In addition to heuristics to direct the search, many rule learning algorithms apply other criteria for filtering out uninteresting rules or for stopping the refinement process at an appropriate point. Stopping and filtering criteria have received considerably less attention in the literature, mostly because their task is more frequently addressed by pruning. The few existing proposals, most notably simple thresholding of the evaluation metric, significance testing, and MDL-based criteria, turn out to be quite diverse approaches to this problem. Although our analysis will show some interesting correspondences between some of these techniques, it is also clear that stopping criteria are far from being well-understood.

The outline of the paper is as follows: We begin with setting up the formal framework of our analysis, most notably the introduction of coverage spaces and isometrics (Section 2), and a brief review of covering algorithms in this context (Section 3). In Sections 4 and 5, we analyze some of the most commonly used evaluation metrics, and show that those with a linear isometric landscape all fit into a common framework. Subsequently, we discuss a few issues related to learning rule sets with the covering algorithm (Section 6). In Section 7, we will look at the most commonly used stopping and filtering criteria. Finally, we discuss a few open questions (Section 8) and related work (Section 9) before we conclude (Section 10). Parts of this paper have previously appeared in Fürnkranz and Flach (2003) and Fürnkranz and Flach (2004).

2. Formal framework

In the following, we define the formal framework in which we perform our analysis. In particular, we will define coverage spaces and discuss their relation to ROC spaces (Section 2.2) and provide formal definitions of the equivalence of evaluation metrics and point out the importance of isometrics in coverage space for identifying such equivalences (Section 2.3).

2.1. Notational conventions

We use capital letters to denote the total number of positive (P) and negative (N) examples in the training set. $p(r)$ denotes the number of true positives and $n(r)$ the number of false positives covered by a rule r. Heuristics are two-dimensional functions of the form $h(n, p)$. We use subscripts to the letter h to differentiate between different heuristics. For brevity

Table 1. Notational conventions for a confusion matrix of positive and negative examples covered or not covered by a rule.

	Covered by rule	Not covered by rule	
Positive example	p	$P - p$	P
Negative example	n	$N - n$	N
	$p + n$	$P + N - p - n$	$P + N$

and readability, we will abridge $h(n(r), p(r))$ with $h(r)$, and omit the argument (r) from functions p, n, and h when it is clear from the context. Table 1 shows a summary of our notational conventions in the form of a four-field confusion matrix.

In the remainder of the paper, we will use the terms *heuristic function* and *evaluation metric* interchangeably (often abbreviated to heuristic and metric).

2.2. *ROC space and coverage space*

In the following, we assume some basic knowledge of ROC analysis as it is typically used for selecting the best classifier for unknown classification costs (Provost & Fawcett, 2001). In brief, ROC analysis compares different classifiers according to their true positive rate (TPR) and false positive rate (FPR), i.e., the percentage p/P of correctly classified positive examples and the percentage n/N of incorrectly classified negative examples. This allows to plot classifiers in a two-dimensional space, one dimension for each of these two measurements. The ideal classifier is at the upper left corner, at point $(1, 0)$, while the origin $(0, 0)$ and the point $(1, 1)$ correspond to the classifiers that classify all examples as negative and positive respectively. If all classifiers are plotted in this ROC space, the best classifier is the one that is closest to $(1, 0)$, where the used distance measure depends on the relative misclassification costs of the two classes. One can also show that all classifiers that are optimal under some (linear) cost model lie on the convex hull of these points in ROC space. Thus, all classifiers that do not lie on this convex hull can be ruled out.[2]

In the particular context of rule learning, we believe it is more convenient to think in terms of absolute numbers of covered examples. Thus, instead of plotting TPR over FPR, we plot the absolute number of covered positive examples over the absolute number of covered negative examples. We call this space *coverage space*.[3]

Definition 2.1 (Coverage space). *Coverage space* is a two-dimensional space of points (n, p), where $0 \leq n \leq N$ denotes the number of negative examples covered by a rule (false positives) and $0 \leq p \leq P$ denotes the number of positive examples covered by a rule (true positives).

Figure 1 shows two examples, one for the case where the number of positive examples P exceeds the number of negative examples N, and one for the opposite case. In all subsequent

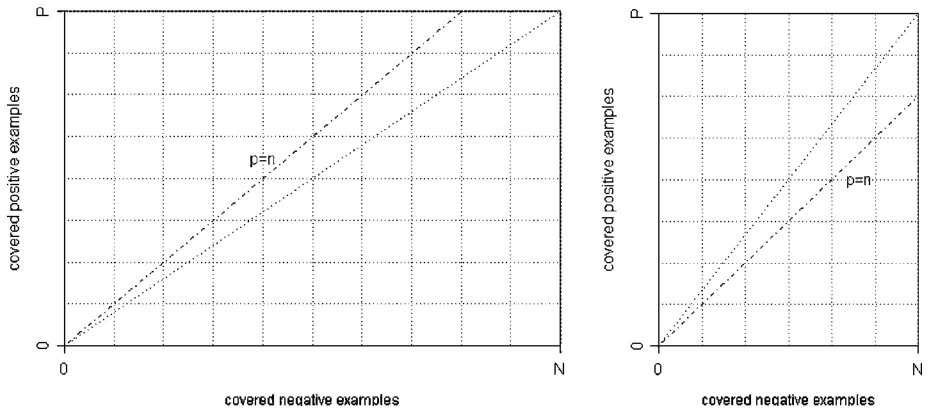

Figure 1. Coverage spaces are ROC spaces based on absolute numbers of covered examples.

figures, we will assume $P \leq N$ (the left graph of figure 1), but this does not affect our results.

A coverage graph can be turned into a ROC graph by simply normalizing the P and N-axes to the scale $[0, 1] \times [0, 1]$. Consequently, the isometrics of a function in a coverage graph can be mapped one-to-one to its isometrics in ROC space. Nevertheless, coverage graphs have several interesting properties that may be of interest depending on the purpose of the visualization. Most notably, the absolute numbers of covered positive and negative examples allow to map the covering strategy into a sequence of nested coverage graphs. This is further discussed in Section 6.1. Table 2 compares some of the properties of coverage spaces to those of ROC spaces.

2.3. *Isometrics and equivalence*

Let us first define some basic properties of rule evaluation metrics.

Table 2. Coverage spaces vs. ROC spaces.

Property	ROC space	Coverage space
x-axis	FPR $= \frac{n}{N}$	n
y-axis	TPR $= \frac{p}{P}$	p
Empty theory R_0	$(0, 0)$	$(0, 0)$
Correct theory R	$(0, 1)$	$(0, P)$
Universal theory $\tilde{R}$	$(1, 1)$	(N, P)
Resolution	$(\frac{1}{N}, \frac{1}{P})$	$(1, 1)$
Slope of diagonal	1	$\frac{P}{N}$
Slope of $p = n$ line	$\frac{N}{P}$	1

Definition 2.2 (Compatible). Two heuristic functions h_1 and h_2 are *compatible* iff for all rules r, s:

$$h_1(r) > h_1(s) \Leftrightarrow h_2(r) > h_2(s).$$

Definition 2.3 (Antagonistic). Two heuristic functions h_1 and h_2 are *antagonistic* iff for all rules r, s:

$$h_1(r) > h_1(s) \Leftrightarrow h_2(r) < h_2(s).$$

Obviously, compatibility and antagonicity are dual concepts:

Lemma 2.4. *h_1 and h_2 are antagonistic iff h_1 and $-h_2$ are compatible.*

Proof: This follows immediately from $h_2(r) > h_2(s) \Leftrightarrow -h_2(r) < -h_2(s)$. □

Note that compatible or antagonistic heuristics have identical regions of equality:

Definition 2.5 (Equality-preserving). Two heuristic functions h_1 and h_2 are *equality-preserving* iff for all rules r, s:

$$h_1(r) = h_1(s) \Leftrightarrow h_2(r) = h_2(s).$$

Theorem 2.6. *Compatible or antagonistic heuristics are equality-preserving.*

Proof: Assume they would not be equality-preserving. This means there exist rules r and s with $h_1(r) = h_1(s)$ but $h_2(r) \neq h_2(s)$. Without loss of generality assume $h_2(r) > h_2(s)$. This implies that $h_1(r) > h_1(s)$ (for compatibility) or $h_1(r) < h_1(s)$ (for antagonicity). Both cases contradict the assumption. □

Note that equality-preserving heuristics are not necessarily compatible or antagonistic, only if we make some straightforward continuity assumptions. Although all subsequently discussed functions are continuous, we note that this does no have to be the case. For all practical purposes, $h(n, p)$ may be regarded as a look-up table that defines a value h for each integer-valued pair (n, p).

Based on the above, we can define the equivalence of heuristics. The basic idea is that heuristic functions are equivalent if they order a set of candidate rules in an identical way. The notion of equivalence should also capture the fact that some learning systems search for minima of their heuristic function, while others search for maxima (e.g., maximizing information gain is equivalent to minimizing entropy).

Definition 2.7 (Equivalence). Two heuristic functions h_1 and h_2 are *equivalent* ($h_1 \sim h_2$) if they are either compatible or antagonistic.

This definition also underlines the importance of isometrics for the analysis of heuristics:

Definition 2.8 (Isometric). An *isometric* of a heuristic h is a line (or curve) in coverage space that connects, for some value c, all points (n, p) for which $h(n, p) = c$.

Equality-preserving heuristics can be recognized by examining their isometrics and establishing that for each isometric line for h_1 there is a corresponding isometric for h_2. Compatible (and antagonistic) heuristics can be recognized by investigating corresponding isometrics and establishing that their associated heuristic values are in the same (the opposite) order.

3. Covering rule learning algorithms

The most popular strategy for learning classification rules is the *covering* or *separate-and-conquer* strategy. It has its roots in the early days of machine learning in the AQ family of rule learning algorithms (Michalski, 1969; Kaufman & Michalski, 2000). It is fairly popular in *propositional rule learning* (cf. CN2 (Clark & Niblett, 1989; Clark & Boswell, 1991), Ripper (Cohen, 1995), or CBA (Liu, Hsu, & Ma, 1998)) as well as in *inductive logic programming (ILP)* (cf. Foil (Quinlan, 1990) and its successors (Džeroski & Bratko, 1992; Fürnkranz, 1994; De Raedt & Van Laer, 1995; Quinlan & Cameron-Jones, 1995) or Progol (Muggleton, 1995)).

In the following, we will briefly recapitulate the key issues of this learning strategy, with a particular focus on its behavior in coverage space. For a detailed survey of this large family of algorithms we refer to Fürnkranz (1999).

3.1. Separate-and-conquer learning

The defining characteristic of this family of algorithms is that they successively learn rules that explain part of the available training examples. Examples that are *covered* by previously learned rules are *separated* from the training set, and the remaining examples are *conquered* by recursively calling the learner to find the next rule.[4] This is repeated until all examples are explained by a rule, or until some other, external *stopping criterion* specifies that the current rule set is satisfactory.

Most algorithms of this family operate in a concept learning framework, i.e., they learn the definition of an unknown concept from positive examples (examples for the concept) and negative examples (counter-examples). Each of the learned rules covers some (as many as possible) of the available positive examples, and possibly some (as few as possible) of the negative examples as well. Thus, the learned rule set describes the hidden concept. For classifying new examples, each rule is tried. If any of the learned rules fires for a given example, the example is classified as positive. If none of them fires, the example is classified as negative. This corresponds to the *closed-world assumption* in the semantics of rule sets (theories) and rules (clauses) in the PROLOG programming language.

3.2. *Covering in coverage space*

Adding a rule to a rule set means that more examples are classified as positive, i.e., it increases the coverage of the rule set. All positive examples that are uniquely covered by the newly added rule contribute to an increase of the true positive rate on the training data. Conversely, covering additional negative examples may be viewed as increasing the false positive rate on the training data. Therefore, adding rule r_{i+1} to rule set R_i effectively moves from a point $R_i = (n_i, p_i)$ (corresponding to the number of negative and positive examples that are covered by previous rules), to a new point $R_{i+1} = (n_{i+1}, p_{i+1})$ (corresponding to the examples covered by the new rule set). Moreover, R_{i+1} will typically be closer to (N, P) and farther away from (0, 0) than R_i.

Consequently, learning a rule set one rule at a time may be viewed as a path through coverage space, where each point on the path corresponds to the addition of a rule to the theory. Such a *coverage path* starts at (0, 0), which corresponds to the empty theory that does not cover any examples. Adding a rule moves to a new point in coverage space, which corresponds to a theory consisting of all rules that have been learned so far. After the final rule has been learned, one can imagine adding yet another rule with a body that is always true. Adding such a rule has the effect that the theory now classifies *all* examples as positive, i.e., it will take us to the point $\tilde{R} = (N, P)$. Figure 2 shows the coverage path for a theory with three rules. Each point R_i represents the rule set consisting of the first i rules.

The final rule set learned by a classifier will typically correspond to the last point on the curve before it reaches (N, P). Note, however, that this need not be the case. This will be briefly discussed in Section 6.3.

3.3. *Greedy specialization*

While the covering loop is essentially the same for all members of the separate-and-conquer family of rule learning algorithms, the individual members differ in the way they learn single rules. Algorithms may use stochastic (Mladenić, 1993) or genetic search (Giordana & Sale, 1992), exhaustive search (Webb, 1995; Muggleton, 1995), or employ association

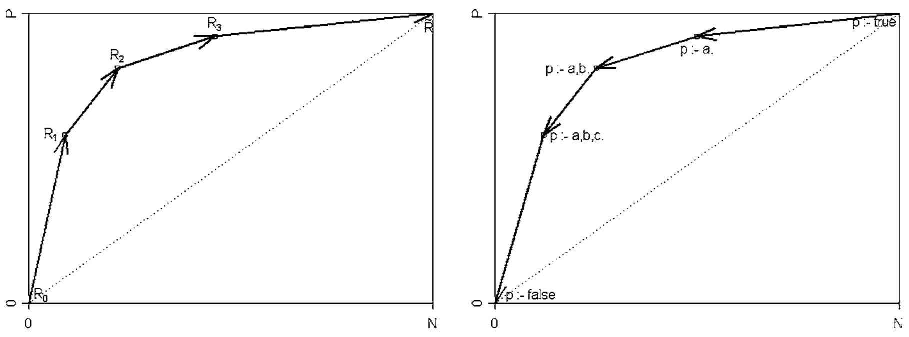

Figure 2. Schematic depiction of the paths through coverage space that are described by (left) the covering strategy of learning a theory by adding one rule at a time and (right) greedy specialization of a single rule.

rule algorithms for finding a candidate set of rules (Liu, Hsu, & Ma, 1998; Jovanoski & Lavrač, 2001). Again, we refer to Fürnkranz (1999) for a survey.

The vast majority of algorithms uses a heuristic top-down hill-climbing or beam search strategy, i.e., they search the space of possible rules by successively specializing the current best rule. Rules are specialized by greedily adding the condition which promises the highest gain according to some *evaluation metric*. Like with adding rules to a rule set, this successive refinement describes a path through coverage space (see figure 2, right). However, in this case, the path starts at the upper right corner (covering all positive and negative examples), and successively proceeds towards the origin (which would be a rule that is too specific to cover any example).

3.4. Overfitting avoidance

In the simplest case, both for refining a rule set and refining a rule, all points on its coverage path are evaluated with an evaluation metric, and the rule set or rule that corresponds to the point with the highest evaluation is selected. However, such an approach may often be too simplistic, in particular because of the danger of *overfitting*. A rule learner typically evaluates a large number of candidate rules, which makes it quite likely that one of them fits the characteristics of the training set by chance. For example, simply using the fraction of positive examples covered by the rule suffers from overfitting because one can always successively refine a rule until it covers only a single positive example and no negative examples. In the worst case, this will happen when all attribute values that describe the example are added to the rule, but quite frequently the resulting rule is much simpler and will still cover a significant portion of the instance space. Such a rule has an optimal value 1, but it is unlikely that it generalizes well to unseen data.

A simple way of countering overfitting is to explicitly specify the region of the coverage space for which it is believed that the provided heuristic evaluations are too weak or too unreliable to be of interest. There are two principal approaches for dealing with this kind of problem: *pre-pruning* algorithms employ additional evaluation metrics for filtering out unpromising rules or for stopping the refinement process, whereas *post-pruning* approaches deliberately learn an overfitting rule or theory and correct it in a post-processing phase. For a discussion of the two alternatives and some proposals for combining them we refer to Fürnkranz (1997). In this paper, we will confine ourselves to pre-pruning heuristics (Section 7).

4. Heuristics with linear isometrics

The ultimate goal of learning is to reach point $(0, P)$ in coverage space, i.e., to learn a correct theory that covers all positive examples, but none of the negative examples. This will rarely ever be achieved in a single step, but a set of rules will be needed to meet this objective. The purpose of a rule evaluation metric is to estimate how close a rule takes you to this ideal point. In the following, we analyze the most commonly used metrics for evaluating the quality of a rule in covering algorithms.

4.1. Basic heuristics

A straightforward strategy for finding a rule that covers some positive and as few negative examples as possible is to minimize the number of covered negative examples for each individual rule (or maximize their negation).

$$h_n = -n$$

This, however, does not capture the intuition that we want to cover as many positive examples as possible. For this, h_p, the number of covered positive examples, could be used.

$$h_p = p$$

Figure 3 shows the isometrics for h_n and h_p, vertical and horizontal lines. All rules that cover the same number of negative (positive) examples are evaluated equally, irrespective of the number of positive (negative) examples they cover.

However, it is trivial to find theories that maximize either of them: h_n is maximal for the empty theory R_0, which does not cover any negative examples, but also no positive examples, and h_p is maximal for the universal theory $\tilde{R}$, which covers all positive and no negative examples. Ideally, one would like to achieve both goals simultaneously.

4.2. Accuracy, weighted relative accuracy, general costs

A straightforward way for trading off covering many positives and excluding many negatives is to simply add up h_n and h_p:

$$h_{\text{acc}} = p - n$$

The isometrics for this function are shown in the left graph of figure 4. Note that the isometrics all have a 45° angle. Thus this heuristic basically optimizes accuracy:

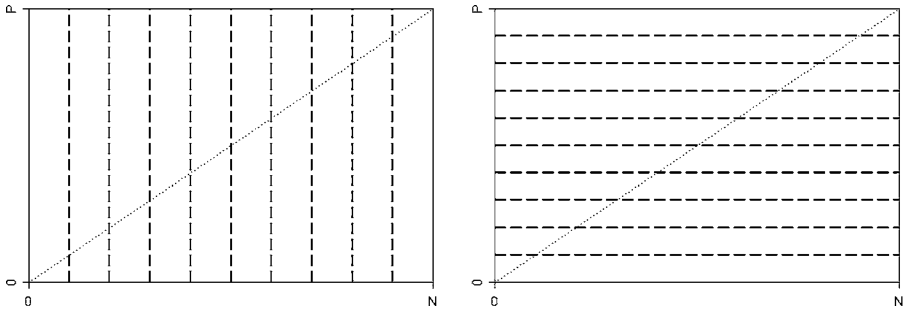

Figure 3. Isometrics for minimizing false positives and for maximizing true positives.

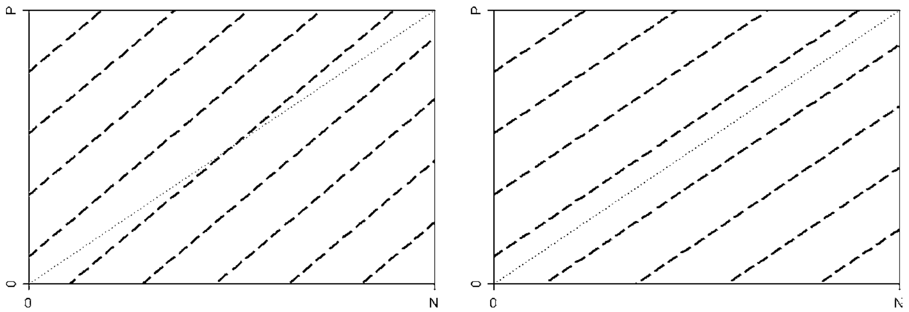

Figure 4. Isometrics for accuracy and weighted relative accuracy.

Theorem 4.1. *h_{acc} is equivalent to accuracy.*

Proof: The accuracy of a theory (which may be a single rule) is the proportion of correctly explained examples, i.e., positive examples that are covered (p) and negative examples that are not covered $(N - n)$, in all examples $(P + N)$. Thus the isometrics are of the form $\frac{p+(N-n)}{P+N} = c$. As P and N are constant, these can be transformed into the isometrics of h_{acc}: $p - n = c_{\text{acc}} = c(P + N) - N$. □

Accuracy has been used as a pruning heuristic in I-REP (Fürnkranz & Widmer, 1994), a modification of h_{acc}, which also subtracts the length of a rule, has been used in PROGOL (Muggleton, 1995).

Optimizing accuracy gives equal weight to covering a single positive example and excluding a single negative example. There are cases where this choice is arbitrary, for example when misclassification costs are not known in advance or when the samples of the two classes are not representative. In such cases, it may be advisable to normalize with the sample sizes:

$$h_{\text{wra}} = \frac{p}{P} - \frac{n}{N} = \text{TPR} - \text{FPR}$$

The isometrics of this heuristic are shown in the right half of figure 4. The main difference with accuracy is that the isometrics are parallel to the diagonal, which reflects that we now give equal weight to increasing the true positive rate (TPR) or to decreasing the false positive rate (FPR). Note that the diagonal encompasses all classifiers that make random predictions.

h_{wra} may be viewed as a simplification of *weighted relative accuracy*. Variations of this heuristic have been well-known in subgroup discovery (Klösgen, 1996), and it has recently been proposed as a rule learning heuristic (Lavrač, Flach, & Zupan, 1999; Todorovski, Flach, & Lavrač, 2000):

Theorem 4.2. *h_{wra} is equivalent to weighted relative accuracy.*

Proof: Weighted relative accuracy is defined as $h_{\text{wra}'} = \frac{p+n}{P+N}(\frac{p}{p+n} - \frac{P}{P+N})$. Using equivalence-preserving transformations (multiplications with constant values like $P + N$), we obtain $h_{\text{wra}'} = \frac{1}{P+N}(p - p\frac{P}{P+N} - n\frac{P}{P+N}) \sim p\frac{N}{P+N} - n\frac{P}{P+N} \sim pN - nP \sim \frac{p}{P} - \frac{n}{N} = h_{\text{wra}}$. □

The two coverage graphs of figure 4 are special cases of a function that allows to incorporate arbitrary cost ratios between false negatives and false positives. The general form of this *linear cost metric* is

$$h_{\text{costs}} = ap - bn \sim (1 - c)p - cn \sim p - dn$$

The parameters a, b, c and d are different ways of specifying the cost trade-off. a denotes the benefit (negative costs) of a true positive, whereas b specifies the costs of a false negative. However, as we are only interested in comparing h_{costs} for different rules, the absolute size of a and b is irrelevant; only their ratio is important for determining the order of the rules. Thus, the benefit of a true positive can be normalized to 1, which results in costs $d = \frac{b}{a}$ for a false positive. $d = 0$ means no costs for false positives (resulting in h_p), whereas $d = \infty$ denotes infinite costs for false positives (i.e., h_n). It is often more convenient to normalize this cost parameter to the range [0, 1], which results in $c = b = \frac{d}{d+1}$, and $1 - c = a = \frac{1}{d+1}$.

In the coverage graph, the cost trade-off is characterized by the slope $d = \frac{b}{a} = \frac{c}{1-c}$ of the parallel isometrics. As a consequence, arbitrary cost ratios can be modeled by h_{costs}. For example, accuracy is characterized by equal costs for false positives and false negatives. Therefore, its isometrics can be obtained with $a = b = d = 1$ or $c = 1/2$. The isometrics of weighted relative accuracy can be obtained by setting $a = 1/P$ and $b = 1/N$ or $c = P/(P + N)$ or $d = P/N$.

4.3. Recall and precision, support and confidence

The most commonly used heuristic for evaluating single rules is to look at the proportion of positive examples in all examples covered by the rule. This metric is known under many different names, e.g., *confidence* in association rule mining, or *precision* in information retrieval. We will use the latter term:

$$h_{\text{pr}} = \frac{p}{p+n}$$

Figure 5 shows the isometrics for this heuristic. Like h_p, precision considers all rules that cover only positive examples to be equally good (the P-axis), and like h_n, it considers all rules that only cover negative examples as equally bad (the N-axis). All other isometrics are obtained by rotation around the origin (0, 0), for which the heuristic value is undefined.

Uses of the pure precision heuristic in rule learning include Pagallo and Haussler (1990) and Weiss and Indurkhya (1991). However, several other, seemingly more complex heuristics can be shown to be equivalent to precision. For example, the heuristic that is used for pruning in Ripper (Cohen, 1995):

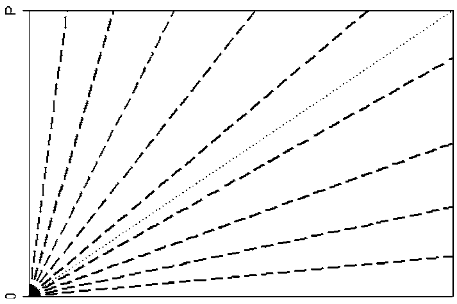

Figure 5. Isometrics for precision.

Theorem 4.3. Ripper*'s pruning heuristic* $h_{rip} = \frac{p-n}{p+n}$ *is equivalent to precision.*

Proof: $h_{rip} = \frac{p}{p+n} - \frac{n}{p+n} = \frac{p}{p+n} - (1 - \frac{p}{p+n}) = 2 * h_{\text{pr}} - 1$ □

In subsequent sections, we will see that more complex heuristics, such as entropy and Gini index, are also essentially equivalent to precision. On the other hand, seemingly minor modifications like the Laplace or m-estimates are not.

Precision and confidence are more frequently used together with their respective counterparts *recall* and *support*.

$$h_{\text{rec}} = \frac{p}{P} = \text{TPR}$$

As P is constant, h_{rec} is obviously equivalent to h_p discussed above. Support is usually defined as the fraction of examples that satisfy both the head and the body of the rule, i.e., $h_{\text{supp}} = \frac{p}{P+N} \sim h_{\text{rec}}$.

4.4. Information content, entropy and gini index

Some algorithms (e.g., PRISM (Cendrowska, 1987)) measure the information content

$$h_{\text{info}} = -\log_2 \frac{p}{p+n}$$

Theorem 4.4. h_{info} *and* h_{pr} *are antagonistic and thus equivalent.*

Proof: $h_{\text{info}} = -\log_2 h_{\text{pr}}$, thus $h_{\text{info}}(r) > h_{\text{info}}(s) \Leftrightarrow h_{\text{pr}}(r) < h_{\text{pr}}(s)$. □

The use of entropy (in the form of information gain) is very common in decision tree learning (Quinlan, 1986), but has also been suggested for rule learning in the original version

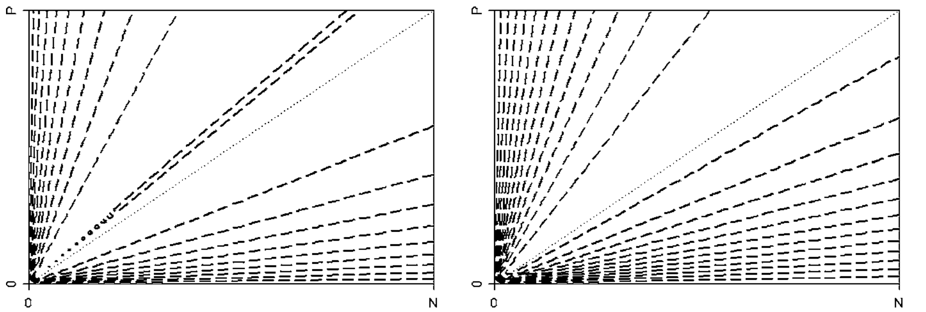

Figure 6. Isometrics for entropy (left) and Gini index (right).

of CN2 (Clark & Niblett, 1989).

$$h_{\text{ent}} = -\left(\frac{p}{p+n}\log_2\frac{p}{p+n} + \frac{n}{p+n}\log_2\frac{n}{p+n}\right)$$

Entropy is not equivalent to information content and precision, even though it seems to have the same isometrics as these heuristics (see figure 6). The difference is that the isometrics of entropy go through the undefined point (0, 0) and continue on the other side of the 45° diagonal. The motivation for this is that the original version of CN2 did not assume a positive class, but labeled its rules with the majority class among the examples covered. Thus rules $r = (n, p)$ and $\bar{r} = (p, n)$ can be considered to be of equal quality because one of them will be used for predicting the positive class and the other for predicting the negative class.

Based on this, we can, however prove the following

Theorem 4.5. *h_{ent} and h_{pr} are antagonistic for $p \geq n$ and compatible for $p \leq n$.*

Proof: $h_{\text{ent}} = -h_{\text{pr}}\log_2 h_{\text{pr}} - (1 - h_{\text{pr}})\log_2(1 - h_{\text{pr}})$ with $h_{\text{pr}} \in [0, 1]$. This function has its maximum at $h_{\text{pr}} = 1/2 \Leftrightarrow p = n$. From the fact that it is strictly monotonically increasing for $p \leq n$, it follows that $h_{\text{pr}}(x) < h_{\text{pr}}(y) \Rightarrow h_{\text{ent}}(x) < h_{\text{ent}}(y)$ in this region. Analogously, $h_{\text{pr}}(x) < h_{\text{pr}}(y) \Rightarrow h_{\text{ent}}(x) > h_{\text{ent}}(y)$ for $p \geq n$, where h_{ent} is monotonically decreasing in h_{pr}. □

We will say that entropy is a *class-neutral* version of precision. In general, a class-neutral heuristic has isometrics that are line-mirrored across a symmetry line, typically the 45° line or the diagonal. A heuristic that is not class-neutral can be made so by re-scaling to $[-1, 1]$ and taking the absolute value. For instance, $|\frac{p-n}{p+n}|$ is a class-neutral version of precision, and therefore (by construction) equivalent to entropy.

In decision tree learning, the Gini index is also a very popular heuristic (Breiman et al., 1984). To our knowledge, it has not been used in rule learning, but we list it for completeness:

$$h_{\text{gini}} = 1 - \left(\frac{p}{p+n}\right)^2 - \left(\frac{n}{p+n}\right)^2 \sim \frac{pn}{(p+n)^2}$$

As can be seen from figure 6, the Gini index has the same isometric landscape as entropy, it only differs in the distribution of the values (hence the lines of the contour plot are a little denser near the axes and less dense near the diagonal). This, however, does not change the ordering of the rules.

Theorem 4.6. *h_{gini} and h_{ent} are equivalent.*

Proof: Like entropy, the Gini index can be formulated in terms of h_{pr} ($h_{\text{gini}} \sim h_{\text{pr}}(1-h_{\text{pr}})$) and both functions have essentially the same shape, i.e., both functions grow or fall with h_{pr} in the same way. □

4.5. Laplace, m-estimate, F- and g-measure

The Laplace and m-estimates (Cestnik, 1990) are very common modifications of h_{pr}, which have, e.g., been used in the second version of CN2 (Clark & Boswell, 1991), m-Foil (Džeroski & Bratko, 1992), and ICL (De Raedt & Van Laer, 1995).

$$h_{\text{lap}} = \frac{p+1}{p+n+2}$$

$$h_m = \frac{p + m\frac{P}{P+N}}{p+n+m}$$

The basic idea of these estimates is to assume that each rule covers a certain number of examples a priori. They compute a precision estimate, but start to count covered positive or negative examples at a number > 0. With the Laplace estimate, both the positive and negative coverage of a rule are initialized with 1 (thus assuming an equal prior distribution), while the m-estimate assumes a prior total coverage of m examples which are distributed according to the distribution of positive and negative examples in the training set.

In the coverage graphs, this modification results in a shift of the origin of the precision isometrics to the point $(-n_m, -p_m)$, where $n_m = p_m = 1$ in the case of the Laplace heuristic, and $p_m = m * P/(P+N)$ and $n_m = m - p_m$ for the m-estimate (see figure 7). The resulting isometric landscape is symmetric around the line that goes through $(-n_m, -p_m)$ and $(0, 0)$. Thus, the Laplace estimate is symmetric around the 45° line, while the m-estimate is symmetric around the diagonal of the coverage graph.

Another noticeable effect of the transformation is that the farther the origin $(-n_m, -p_m)$ moves away from $(0, 0)$, the more the isometrics in the relevant window $(0, 0) - (N, P)$ approach parallel lines. For example, the isometrics of the m-estimate converge towards the isometrics of weighted relative accuracy for $m \to \infty$ (see Theorem 4.8 below).

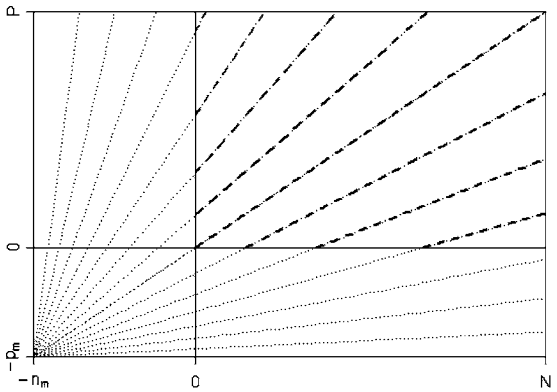

Figure 7. Isometrics for the m-estimate.

This is maybe best illustrated if we set $p_m = 0$, i.e., if we assume the rotation point to be on the N-axis. We will call this the g-measure:

$$h_g = \frac{p}{p + n + g}$$

Figure 8 shows how for $g \to \infty$, the slopes of h_g's isometrics becomes flatter and flatter and converge towards the axis-parallel lines of h_{rec}. On the other hand, for $g = 0$, the measure is obviously equivalent to precision. Thus, the g-measure may be regarded as a simple way for trading off precision and recall, or support and confidence.

In fact, a number of equivalent measures were considered by other authors. Flach (2003) showed that for $g = P$, h_g is equivalent to the F-measure, a frequently used trade-off between recall and precision (van Rijsbergen, 1979). Flach also proposed the G-measure ($h_G = \frac{p}{n+P}$) as an equivalent simplification of the F-measure. This, in turn, is a special case of the heuristic $\frac{p}{n+g}$ that was proposed by Gamberger and Lavrač (2002) for subgroup discovery. We demonstrate the equivalence of the latter heuristic with our g-measure.

Theorem 4.7. *For* $g \geq 0$, $h_g \sim \frac{p}{n+g}$

Proof: We show equivalence via compatibility.

$$\begin{aligned}
h_g(r_1) > h_g(r_2) &\Leftrightarrow \frac{p_1}{p_1 + n_1 + g} > \frac{p_2}{p_2 + n_2 + g} \\
&\Leftrightarrow p_1(p_2 + n_2 + g) > p_2(p_1 + n_1 + g) \\
&\Leftrightarrow p_1 p_2 + p_1(n_2 + g) > p_2 p_1 + p_2(n_1 + g) \\
&\Leftrightarrow p_1(n_2 + g) > p_2(n_1 + g) \\
&\Leftrightarrow \frac{p_1}{n_1 + g} > \frac{p_2}{n_2 + g}
\end{aligned}$$

□

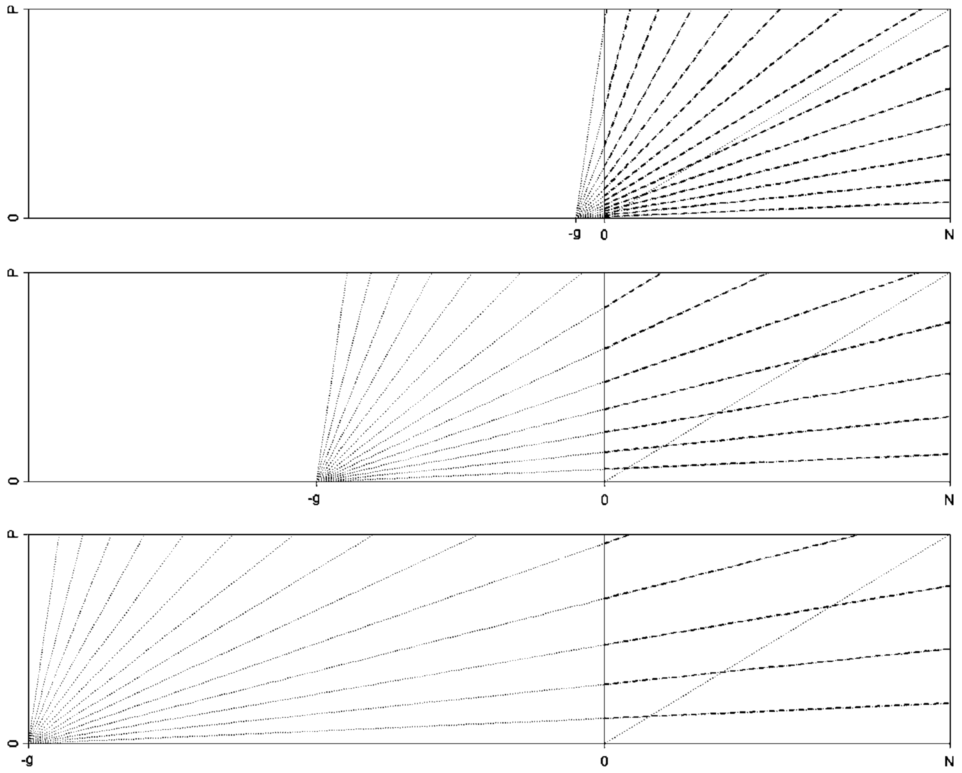

Figure 8. Isometrics of h_g for increasing values of g.

4.6. The generalized m-estimate

The above discussion leads us to the following straightforward generalization of the m-estimate, which takes the rotation point of the precision isometrics as a parameter:

$$h_{\mathrm{gm}} = \frac{p + mc}{p + n + m} = \frac{p + a}{(p + a) + (n + b)}$$

The second version of the heuristic basically defines the rotation point by specifying its co-ordinates $(-b, -a)$ in coverage space $(a, b \in [0, \infty])$. The first version uses m as a measure of how far from the origin the rotation point lies using the sum of the co-ordinates as a distance measure. Hence, all points with distance m lie on the line that connects $(0, -m)$ with $(-m, 0)$, and c specifies where on this line the rotation point lies. For example, $c = 0$ leaves the positive co-ordinate unchanged and moves the origin to $(-m, 0)$, whereas $c = 1$ denotes $(0, -m)$ as the rotation point. The line that connects the rotation point and $(0, 0)$ has a slope of $\frac{c}{1-c}$. Obviously, both versions of h_{gm} can be transformed into each other by choosing $m = a + b$ and $c = \frac{a}{a+b}$ or $a = mc$ and $b = m(1 - c)$.

Theorem 4.8. *For $m = 0$, h_{gm} is equivalent to h_{pr}, while for $m \to \infty$, its isometrics converge to h_{costs}.*

Proof: $m = 0$: trivial.

$m \to \infty$: By construction, an isometric of h_{gm} through the point (n, p) connects this point with the rotation point $(-(1-c)m, -cm)$ and has the slope $\frac{p+cm}{n+(1-c)m}$. For $m \to \infty$, this slope converges to $\frac{c}{1-c}$ for all points (n, p). Thus all isometrics converge towards parallel lines with the slope $\frac{c}{1-c}$. □

Theorem 4.8 shows that h_{gm} may be considered as a general model of heuristic functions with linear isometrics that has two parameters: $c \in [0, 1]$ for trading off the misclassification costs between the two classes, and $m \in [0, \infty]$ for trading off between precision h_{pr} and the linear cost metric h_{costs}.[5] Therefore, all heuristics discussed in this section may be viewed as equivalent to some instantiation of this general model.

5. Heuristics with non-linear isometrics

All heuristics considered so far have linear isometrics. This is a reasonable assumption as concepts are typically evaluated with linear cost metrics (e.g., accuracy or cost matrices). However, these evaluation metrics are concerned with evaluating complete rules. As the value of an incomplete rule lies not in its ability to discriminate between positive and negative examples, but in its potential of being *refinable* into a high-quality rule, it might well be the case that different types of heuristics are useful for evaluating incomplete candidate rules. One could, for example, argue that for candidate rules that cover many positive examples it is less important to exclude negatives than for rules with low coverage, because high-coverage candidates may still be refined accordingly. Similarly, overfitting could be countered by penalizing regions with low coverage. Possibly, these and similar problems could be better addressed with a non-linear isometric landscape, even though the learned rules will eventually be used under linear cost models.

In this section, we will look at two non-linear information metrics: Foil's information gain, and the correlation heuristic used in Fossil.

5.1. Foil*'s information gain*

Foil's version of information gain (Quinlan, 1990), unlike ID3's and C4.5's version (Quinlan, 1986), is tailored to rule learning, where one only needs to optimize one successor branch as opposed to multiple successor nodes in decision tree learning. It differs from the heuristics mentioned so far in that it does not evaluate an entire rule, but only the effect of specializing a rule by adding a condition. More precisely, it computes the difference in information content of the current rule and its predecessor r', weighted by the number of covered positive examples (as a bias for generality). The exact formula is[6]

$$h_{\text{foil}} = p\left(\log_2 \frac{p}{p+n} - \log_2 c\right)$$

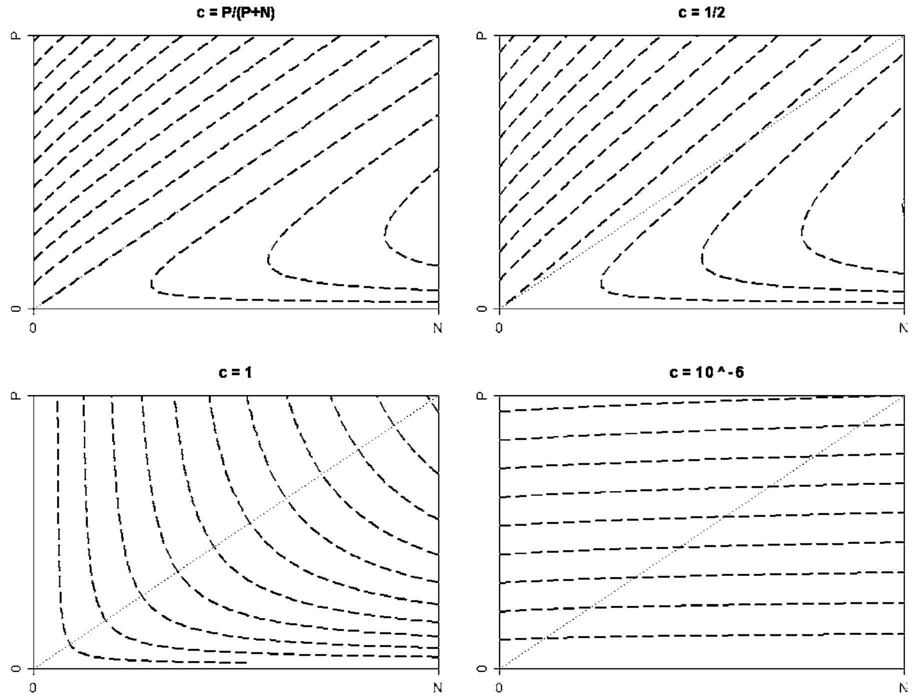

Figure 9. Isometrics for information gain as used in Foil. The curves show different values c for the precision of the parent rule.

where $c = h_{\text{pr}}(r')$ is the precision of the parent rule. In the following, it will turn out to be convenient to interpret c as a cost parameter taking values in the interval [0, 1].

Figure 9 shows the isometrics of h_{foil} for four different settings of c. Although the isometrics are non-linear, they appear to be linear in the region above the isometric that goes through (0, 0). Note that this isometric, which we will call the *base line*, has a slope of $\frac{c}{1-c}$: In the first graph ($c = \frac{P}{P+N}$) it is the diagonal, in the second graph ($c = 1/2$) it has a 45° slope, and in the lower two graphs ($c = 1$ and $c = 10^{-6}$) it coincides with the vertical and horizontal axes respectively.

Note that the base line represents the classification performance of the parent rule. In the area below the base line are the cases where the precision of the rule is smaller than the precision of its predecessor. Such a refinement of a rule is usually not considered to be relevant. In fact, this is also the region where the information gain is negative, i.e., an information loss. The points on the base line have information gain 0, and all points above it have a positive gain.

For this reason, we will focus our analysis on the region above the base line, which is the area where the refined rule improves upon its parent. In this region, the isometrics appear to be linear and parallel to the base line, just like the isometrics of h_{costs}. In fact, in Fürnkranz and Flach (2003), we conjectured that in this part of the coverage space, h_{foil} is equivalent to h_{costs}. However, closer investigation reveals that this is not the case. The farther the isometrics move away from the base line, the steeper they get. Also, they are not exactly linear but slightly bent towards the P-axis, i.e., purer rules are preferred. Nevertheless, it is obvious that both heuristics are closely related, and in fact we can show the following:

Theorem 5.1. *Let $c = h_{pr}(r')$ be the precision of r', the predecessor of rule r. In the relevant region $\frac{p}{p+n} > c$, $h_{costs}(r)$ is equivalent to the first-order Taylor approximation of $h_{foil}(r)$.*

Proof:

$$h_{\text{foil}} = p\left(\log_2 \frac{p}{p+n} - \log_2 c\right) = -p \log_2 \frac{c(p+n)}{p} \sim -p \ln \frac{c(p+n)}{p}$$

The Taylor expansion for $\ln(1+x)$ converges for $-1 < x \leq 1$, and the first-degree approximation is $\ln(1+x) \approx x$. Recall that the interesting region of h_{foil} is where $\frac{p}{p+n} \geq c$, i.e., where the precision of the current rule r exceeds the precision of the parent rule r' (otherwise there would be no point in refining r' to r). In this region, $0 < c \leq c/\frac{p}{p+n} = \frac{c(p+n)}{p} \leq 1$. Thus

$$\begin{aligned} h_{\text{foil}} &\sim -p \ln \frac{c(p+n)}{p} = -p \ln\left(1 + \left(\frac{c(p+n)}{p} - 1\right)\right) \\ &\approx -p\left(\frac{c(p+n)}{p} - 1\right) = -p\frac{c(p+n) - p}{p} = p - c(p+n) \\ &= (1-c)p - cn \end{aligned}$$

□

It should be pointed out that Theorem 5.1 should not be interpreted in the way that h_{costs} is a good approximation for h_{foil}. In fact, while the first-order Taylor approximation of $\ln 1 + x$ is quite good near $x = 0$, it can become arbitrarily bad for $x \to -1$. In our case, we can expect a good approximation if c is close to 1 and a bad approximation if c approaches 0 because $0 < c \leq x + 1 = c(p+n)/p \leq 1$.

On the other hand, it is not strictly necessary to have a good approximation for maintaining the isometric landscape of a function, as we have seen on several examples for equivalent heuristics in the previous section. There is a difference between one heuristic approximating another, and its isometrics approximating the other's. A full analysis of Foil's information gain requires a fuller understanding of what it means to approximate an isometric landscape, which we leave for future work.

5.2. *Correlation and χ^2 statistic*

The correlation metric has been proposed by Fürnkranz (1994) for the rule learner Fossil, a variant of Foil. It computes the correlation coefficient between the predictions made by the

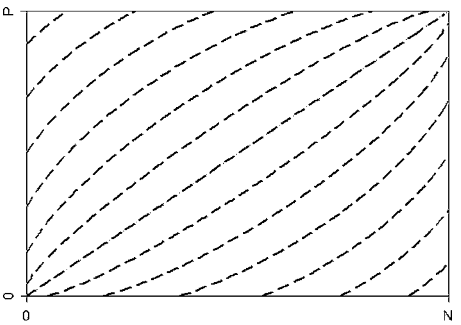

Figure 10. Isometrics for the four-field correlation coefficient.

rule and the true labels from a four-field confusion matrix. The formula is

$$h_{\text{corr}} = \frac{p(N-n)-(P-p)n}{\sqrt{PN(p+n)(P-p+N-n)}} = \frac{pN-Pn}{\sqrt{PN(p+n)(P-p+N-n)}}$$

Recall that p and $N-n$ denote the fields on the diagonal of the confusion matrix (the true positives and the true negatives), whereas n and $P-p$ denote the false positives and false negatives respectively. The terms in the denominator are the sums of the rows and columns of the confusion matrix.

Figure 10 shows the isometric plot of the correlation metric. The isometrics are clearly non-linear and symmetric around the base line on the diagonal (rules that have correlation 0). This symmetry around the diagonal is also characteristic of the linear weighted relative accuracy heuristic h_{wra}, so it is interesting to compare the properties of these two metrics.

Above that base line, the isometrics are bent towards the P and N axes respectively. Thus, h_{corr} has a tendency to prefer purer rules (covering fewer negative examples) and more complete rules (covering more positive examples).

Moreover, the linear isometrics that result from connecting the points where the isometrics intersect with the $n = 0$ axis (on the left) and the $p = P$ axis (on the top), are *not* parallel to the diagonal (as they are for h_{wra}). In fact, it can be shown that the slope of these lines is $\frac{1-\bar{p}/N}{1-\bar{p}/P}$ where $(0, \bar{p})$ is the intersection of the isometric with the P-axis. Obviously, these lines are only parallel for a class balanced problem ($P = N$). For $P < N$, the slope of the isometrics will increase for increasing $\bar{p}$ and approach infinity for $\bar{p} \to P$. Thus, the closer a rule gets to the target point $(0, P)$, the more important it will become to exclude remaining covered negative examples, and the less important to cover additional positive examples.

For example, the rule $(0, P-1)$ that covers all but one positive examples and no negatives and the rule $(1, P)$ that covers all positive examples and one negative example have identical evaluations. On the other hand, the rule $(0, 1)$ covering one positive and no negatives has an evaluation proportional to $1/P$ and is therefore preferred over the rule $(N-1, P)$ that covers all positives but excludes only a single negative example (the rule has an evaluation proportional to $1/N$). Keep in mind that we assumed $P < N$ here; the opposite preference would be the case for $P > N$.

In summary, the correlation metric has a tendency to prefer purer or more complete rules. Moreover, for rules that are far from the target $(0, P)$, it gives higher preference to handling the minority class (i.e., cover positive examples for $P < N$ and exclude negative examples for $N < P$), whereas the priorities for these two objectives become more and more equal the closer a rule is to the target.[7]

Note that the four-field correlation coefficient is basically a normalized version of the χ^2 statistic over the four events in the table (see, e.g., Mittenecker, 1983, or other textbooks on applied statistics).

Theorem 5.2. $h_{chi^2} = (P + N)\, h^2_{corr}$.

Proof: The χ^2 statistic for a four-field confusion matrix (cf. Table 1) is the sum of the squared differences between the expected and the observed values of the four fields in the matrix, divided by the expected values.

The expected values for the four fields in the 2×2 confusion matrix are:

$$\begin{pmatrix} \frac{P(p+n)}{P+N} & \frac{N(p+n)}{P+N} \\ \frac{P(P+N-p-n)}{P+N} & \frac{N(P+N-p-n)}{P+N} \end{pmatrix}$$

Subtracting this from the actual values (Table 1):

$$\begin{pmatrix} p - \frac{P(p+n)}{P+N} & n - \frac{N(p+n)}{P+N} \\ P - p - \frac{P(P+N-p-n)}{P+N} & N - n - \frac{N(P+N-p-n)}{P+N} \end{pmatrix}$$
$$= \begin{pmatrix} \frac{p(P+N)-P(p+n)}{P+N} & \frac{n(P+N)-N(p+n)}{P+N} \\ \frac{(P-p)(P+N)-P(P+N-p-n)}{P+N} & \frac{(N-n)(P+N)-N(P+N-p-n)}{P+N} \end{pmatrix}$$
$$= \frac{1}{P+N}\begin{pmatrix} pN - nP & nP - pN \\ nP - pN & pN - nP \end{pmatrix} = \frac{pN - Pn}{P+N}\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$$

Squaring these differences, dividing them by the expected values, and summing up the four fields yields:

$$h_{\mathrm{chi}^2} = \frac{(pN - Pn)^2}{(P+N)^2}\left(\frac{P+N}{P(p+n)} + \frac{P+N}{N(p+n)} + \frac{P+N}{P(P+N-p-n)} + \frac{P+N}{N(P+N-p-n)}\right)$$

$$= \frac{(pN-Pn)^2}{P+N}\,\frac{N(P+N-p-n)+P(P+N-p-n)+N(p+n)+P(p+n)}{PN(p+n)(P+N-p-n)}$$
$$= \frac{(pN-Pn)^2}{P+N}\,\frac{(P+N)^2}{PN(p+n)(P+N-p-n)} = (P+N)h^2_{\text{corr}}$$

□

Thus, h_{chi^2} may be viewed as a class-neutral version of h_{corr}. Its behavior has been previously analyzed in ROC space. Bradley (1996) argued that the non-linear isometrics of the χ^2-statistic helps to discriminate classifiers that do "little work" from classifiers that achieve the same accuracy (or cost line) but are preferable in terms of other metrics like sensitivity and specificity. The above-mentioned paper also shows how h_{chi^2} can be modified to incorporate other cost models.

Finally, we draw attention to the fact that the Gini splitting criterion, defined as impurity decrease from parent to children where impurity is defined by the Gini index (see Section 4.4), is equivalent to χ^2. Here, we restrict attention to binary splits and two classes: in this case, (N, P) can be interpreted as the coverage of the unsplit parent, and (n, p) and $(N-n, P-p)$ denote the coverage of the two children. Gini-split is then defined as

$$h_{\text{gini-split}} = \frac{PN}{(P+N)^2} - \frac{p+n}{P+N}\frac{pn}{(p+n)^2} - \frac{P-p+N-n}{P+N}\frac{(P-p)(N-n)}{(P-p+N-n)^2}$$

where the first term is the Gini index of the parent, and the second and third terms are the Gini index of the children, weighted with their coverage. This expression can be simplified to

$$h_{\text{gini-split}} = \frac{1}{P+N}\left[\frac{PN}{P+N} - \frac{pn}{p+n} - \frac{(P-p)(N-n)}{P-p+N-n}\right] \tag{1}$$

Theorem 5.3. *$h_{\textit{gini-split}} \sim h_{\textit{chi}^2}$.*

Proof: The expression inside the brackets in Eq. (1) can be rewritten as a fraction with denominator $(P+N)(p+n)(P-p+N-n)$ and enumerator

$$\begin{aligned}
&PN(p+n)(P-p+N-n) - pn(P+N)(P-p+N-n)\\
&\quad -(P-p)(N-n)(P+N)(p+n) = P^2Np - PNp^2 + PN^2p - PNpn\\
&\quad + P^2Nn - PNpn + PN^2n - PNn^2 - P^2pn + Pp^2n - \underline{PNpn} + Ppn^2\\
&\quad - \underline{PNpn} + Np^2n - N^2pn + Npn^2 - P^2Np - P^2Nn - PN^2p - PN^2n\\
&\quad + P^2pn + \underline{P^2n^2} + PNpn + PNn^2 + PNp^2 + PNpn + \underline{N^2p^2} + N^2pn\\
&\quad - Pp^2n - Ppn^2 - Np^2n - Npn^2
\end{aligned}$$

All terms but the underlined ones vanish, so we have

$$
\begin{aligned}
h_{\text{gini-split}} &= \frac{1}{P+N}\frac{(pN-Pn)^2}{(P+N)(p+n)(P-p+N-n)} \\
&= \frac{PN}{(P+N)^2}\frac{(pN-Pn)^2}{PN(p+n)(P-p+N-n)} = \frac{PN}{(P+N)^3}h_{\text{chi}^2}
\end{aligned}
$$

□

We conjecture that this result can be generalized to non-binary splits and more than two classes, but this is currently left as an open problem.

6. Evaluation of rule sets

So far, we have been considering heuristics for evaluating single rules, which map to points in coverage space. In this section we consider the evaluation of sets of rules. We show that the sequence of training sets and intermediate hypotheses can be viewed as a trajectory through coverage space, resulting in a set of nested coverage spaces (Section 6.1). We then proceed by investigating the behavior of precision and the linear cost metric in this context, and show that precision aims at (locally) optimizing the area under the curve (ignoring costs), while the cost metric tries to directly find a (global) optimum under known or assumed costs (Section 6.2). Finally, we will briefly discuss the effect of re-ordering the rules in a rule set (Section 6.3).

6.1. Covering and nested coverage spaces

Of particular interest for the covering approach is the property that coverage graphs reflect a change in the total number or proportion of positive (P) and negative (N) training examples via a corresponding change in the relative sizes of the P and N-axes. ROC analysis, on the other hand, would rescale the new dimensions to the range [0, 1], which has the effect of changing the slope of all lines that depend on the relative sizes of p and n. As a consequence, the coverage graph for a subset of a training set can be drawn directly into the coverage graph of the entire set. In particular, the sequence of training sets that are produced by the recursive calls of the covering strategy—after each new rule all covered examples are removed and the learner calls itself on the remaining examples—can be visualized by a nested sequence of coverage graphs.

This is illustrated in figure 11, which shows a nested sequence of coverage spaces. Each space PN_i has its origin in the point $R_i = (n_i, p_i)$, which corresponds to the theory learned so far. Thus, its learning task consists of learning a theory for the remaining $P - p_i$ positive and $N - n_i$ negative examples that are not yet covered by theory R_i. Each new rule will cover a few additional examples and thus reduce the corresponding coverage space.

Note that some separate-and-conquer algorithms only remove the covered positive examples, but not the covered negative examples. This situation corresponds to reducing only

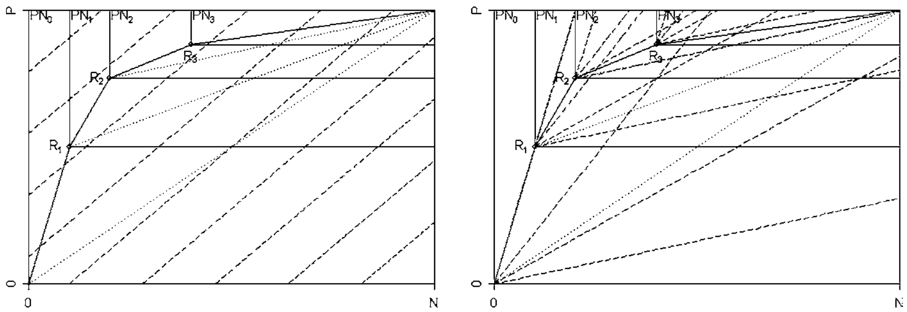

Figure 11. Isometrics for accuracy and precision in nested coverage spaces.

the height of the coverage graph, but not its width. Obviously, this breaks the nesting property because rules are locally evaluated in a coverage graph with full width (all negative examples), but for getting their global evaluation in the context of previously learned rules, all previously covered negative examples have to be removed. While we do not expect a big practical difference between both approaches, removing all examples seems to be the conceptually cleaner solution.

6.2. *Global and local optima*

The nesting property implies that evaluating a single rule in the reduced dataset $(N - n_i, P - p_i)$ amounts to evaluating the point in the subspace PN_i that has the point (n_i, p_i) as its origin. Moreover, the rule that maximizes the chosen evaluation function in PN_i is also a maximum point in the full coverage space if the the evaluation metric does not depend on the size or the shape of the coverage graph.

The linear cost metric h_{costs} is such a heuristic: a local optimum in the subspace PN_i is also optimal in the global coverage space because all isometrics are parallel lines with the same angle, and nested coverage spaces (unlike nested ROC spaces) leave angles invariant. Precision, on the other hand, cannot be nested in this way. The evaluation of a given rule depends on its location relative to the origin of the current subspace PN_i.

This is illustrated in figure 11. The subspaces PN_i correspond to the situation after removing all examples covered by the rule set R_i. The left graph of figure 11 shows the case for accuracy: the accuracy isometrics are all parallel lines with a 45° slope. The right graph shows the situation for precision: each subspace PN_i evaluates the rules relative to its origin, i.e., h_{pr} always rotates around (n_i, p_i) in the full coverage space, which is (0,0) in the local space. Thus, global optimization for precision (or for h_{gm} in general) could be realized by making the generalized m-measure h_{gm} adaptive by choosing $a = p_i$ and $b = n_i$ in subspace PN_i.

However, local optimization is not necessarily bad. At each point (n, p), the slope of the line connecting (0, 0) with (n, p) is p/n.[8] h_{pr} picks the rule that promises the steepest ascent from the origin. Thus, if we assume that the origin of the current subspace is a point of a ROC curve (or a coverage path), we may interpret h_{pr} as making a locally optimal choice for continuing a ROC curve. The choice is optimal in the sense that picking the rule with the steepest ascent locally maximizes the area under the ROC curve.

However, if the cost model is known, this choice may not necessarily be globally optimal. For example, if the point R_2 in figure 11 could be reached in one step, h_{acc} would directly go there because it has the better global value under the chosen cost model (accuracy), whereas h_{pr} would nevertheless first learn R_1 because it promises a greater area under the ROC curve.

In brief we may say that h_{pr} aims at optimizing under unknown costs by (locally) maximizing the area under the ROC curve, whereas h_{costs} tries to directly find a (global) optimum under known (or assumed) costs.

6.3. Reordering rule sets

In the previous sections we have seen that learning a set of rules is a path through coverage space starting at the point $R_0 = (0, 0)$ (the theory covering no examples) and ending at $\tilde{R} = (N, P)$ (the theory covering all examples). Each intermediate rule set R_i is a potential classifier, and one of them has to be selected as the final classifier. However, the curve described by the learner will frequently not be convex. The left graph of figure 12 shows a case where the rule set R_2 consisting of rules $\{r_1, r_2\}$ is not on the convex hull of the classifiers. Thus, only $R_1 = \{r_1\}$ and $R_2 = \{r_1, r_2, r_3\}$ are potential candidates for the final classifier.

Interestingly, in some cases it may be quite easy to obtain better classifiers by simply re-ordering the rules. The right graph of figure 12 shows the best result that can be obtained by swapping the order of rules r_2 and r_3 resulting in the classifier $R_2' = \{r_1, r_3\}$. Note, however, that this graph is drawn under the assumption that rules r_2 and r_3 cover disjoint sets of examples, in which case the quadrangle $R_1 - R_2 - R_3 - R_2'$ is a parallelogram. In general, however, r_3 may cover some of the examples that were previously already covered by r_2, which may change the shape of the quadrangle considerably, depending on whether the majority of the overlap are positive or negative examples. Thus, it is not guaranteed that

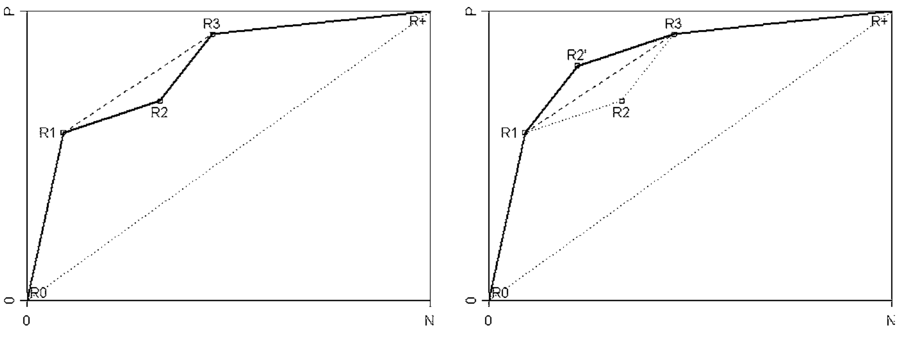

Figure 12. A concavity in the path of a rule learner (left), and the result of a successful fix by swapping the order of the second and third rule (right).

swapping of the rules will indeed increase the area under the ROC curve or make the curve convex.

In this context it is interesting to make a connection to the work by Ferri, Flach, and Hernández (2002). They proposed a method to relabel decision trees in the following way. First, all k leaves (or branches) are sorted in decreasing order according to h_{pr}. Then, a valid labeling is obtained by splitting the ordering in two, and labeling all leaves before the split point positive and the rest negative. This results in $k + 1$ labelings, and the corresponding ROC curve is necessarily convex. An optimal point can be chosen in the usual way, once the cost model is known. In our context, ordering branches corresponds to ordering rules, and relabeling corresponds to picking the right subset of the learned rules: labeling a rule as positive corresponds to including the rule in the theory, while labeling a rule as negative effectively deletes the rule because it will be subsumed by the final default rule that always predicts the negative class. However, the chosen subset is only guaranteed to be optimal if the rules are mutually exclusive. Ferri, Flach, and Hernández (2002) use this technique to derive a novel splitting criterion for decision tree learning. Presumably, a similar criterion could be derived for rule learning, which we regard as a promising topic for future work.

Finally, we note that Flach and Wu (2003) have addressed the problem of repairing concavities in ROC curves in the context of the naive Bayesian classifier. We are currently investigating whether similar techniques are applicable to rule learners.

7. Stopping and filtering criteria

In addition to their regular evaluation metric, many rule learning algorithms employ separate criteria to filter out uninteresting candidates and/or to fight overfitting. There are two slightly different approaches: *stopping criteria* determine when the refinement process should stop and the current best candidate should be returned, whereas *filtering criteria* determine regions of acceptable performance.

Both concepts are closely related. In particular, filtering criteria are often also used as stopping criteria: If no further rule can be found within the acceptable region of a filtering criterion, the learned theory is considered to be complete. Basically the same technique is also used for refining single rules: if no refinement is in the acceptable region, the rule is considered to be complete, and the specialization process stops. For this reason, we will often use the term stopping criterion instead of filtering criterion because this is the more established terminology.

The most popular approach is to impose a threshold upon one or more evaluation metrics. Thus this type of filtering criterion consists of a heuristic function and a threshold value that specifies that only rules that have an evaluation above this value are of interest. The threshold may be chosen by the user or be automatically adapted to certain characteristics of the rule (e.g., its encoding length).

In the following, we analyze a few popular filtering and stopping criteria for greedy specialization: minimum coverage constraints, support and confidence, significance tests, encoding length restrictions, and Fossil's correlation cutoff. We use coverage space to analyze stopping criteria, by visualizing regions of acceptable hypotheses.

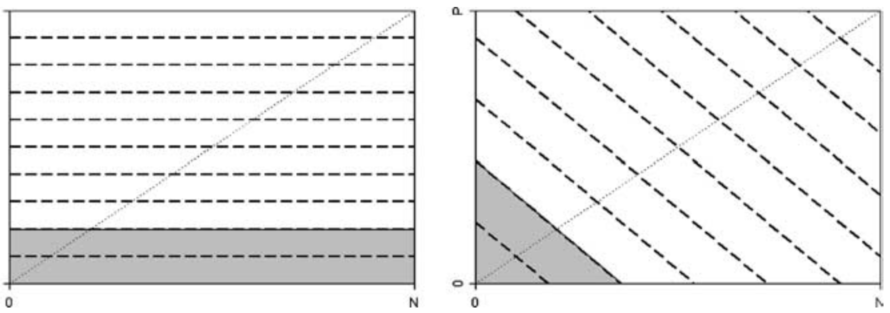

Figure 13. Thresholds on minimum coverage of positive examples (left) and total number of examples (right).

7.1. Minimum coverage constraints

The simplest form of overfitting avoidance is to disregard rules with low coverage. For example, one could require that a rule covers a certain minimum number of examples or a minimum number of positive examples. These two cases are illustrated in figure 13. The graph on the left shows the requirement that a minimum fraction (here 20%) of the positive examples in the training set are covered by the rule. All rules in the gray area are thus excluded from consideration. The right graph illustrates the case where a minimum fraction (here 20%) of examples needs to be covered by the rule, regardless of whether they are positive or negative. Changing the size of the fraction will cut out different slices of the coverage space, each delimited with a coverage isometric (−45° lines). Clearly, in both cases the goal is to fight overfitting by filtering out rules whose quality cannot reliably estimated because of the small number of training examples they cover. Notice that we can include a cost model in calculating the coverage of positive and negative examples, thereby changing the slope of the coverage isometrics. The two cases in figure 13 are then special cases of this general cost-based coverage metric.

7.2. Support and confidence

There is no reason why a single measure should be used for filtering out unpromising rules. The most prominent example for combining multiple estimates are the thresholds on support and confidence that are used mostly in association rule mining algorithms, but also in classification algorithms that obtain the candidate rules for the covering loop in an association rule learning framework (Liu, Hsu, & Ma, 1998; Liu, Ma, & Wong, 2000; Jovanoski & Lavrač, 2001).

Figure 14 illustrates the effect of thresholds on support and confidence in coverage space. Together, they specify an area for valid rules around the (0, P)-point in coverage space. Rules in the grey areas will be filtered out. The dark grey region shows a less restrictive combination of the two thresholds, the light grey region a more restrictive setting. In effect, confidence constrains the quality of the rules, whereas support aims at ensuring a minimum reliability by filtering out rules whose confidence estimate originates from too few positive

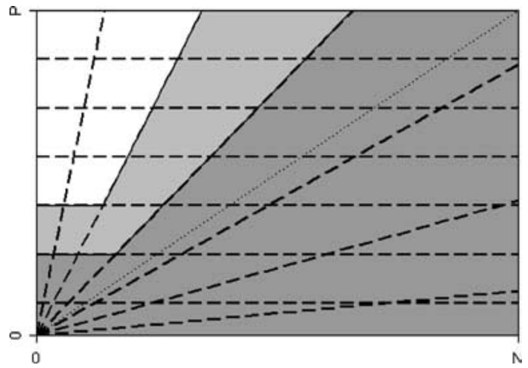

Figure 14. Filtering rules with minimum support and minimum confidence.

examples. The combined effect is quite similar to a threshold on the g- or F-measures. (Section 4.5), but it allows for more rules in the region near the intersection of the two threshold lines.

7.3. CN2*'s significance test*

CN2 (Clark & Niblett, 1989; Clark & Boswell, 1991) filters rules for which the distribution of the covered examples is not statistically significantly different from the distribution of examples in the full data set. To this end, it computes the *likelihood ratio statistic*:

$$h_{\mathrm{lrs}} = 2\left(p \log \frac{p}{e_p} + n \log \frac{n}{e_n}\right)$$

where $e_p = (p+n)\frac{P}{P+N}$ and $e_n = (p+n)\frac{N}{P+N} = (p+n) - e_p$ are the number of positive and negative examples one could expect if the $p + n$ examples covered by the rule were distributed in the same way as the $P + N$ examples in the full data set.

Figure 15 illustrates CN2's filtering criterion. The dark grey area shows the location of the rules that will be filtered out because it can not be established with 95% confidence that the class distribution of the covered examples is different from the class distribution in the full dataset. The light grey area shows the set of rules that will be filtered out if 99% confidence is required. Obviously, the area is symmetric around the diagonal, which corresponds to random guessing.

Other significance tests than the likelihood ratio could be used. For instance, we have already discussed h_{chi^2} in Section 5.2. In the case of a two-by-two contingency table, both h_{chi^2} and h_{lrs} are distributed as χ^2 with one degree of freedom, but that doesn't mean that they are equivalent as heuristics. In fact, there are some interesting differences between the isometric landscapes in figures 10 and 15. More specifically, the likelihood ratio isometrics in figure 15 do not depend on the size of the training set. Any graph corresponding to a bigger training set with the same class distribution (a proportion of $P/(P + N)$ positive

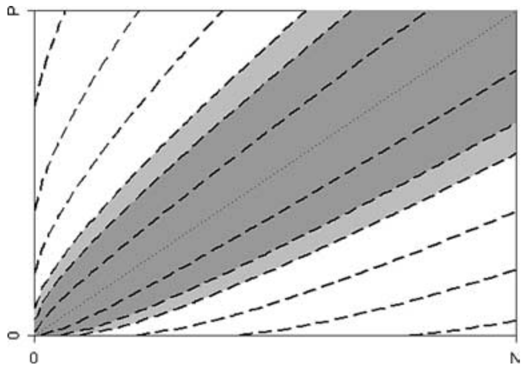

Figure 15. Illustration of CN2's significance test. Shown are the regions that would not pass a 95% (dark grey) and a 99% (light grey) significant test.

examples) will contain this graph in its lower left corner. In other words, whether a rule covering p positive and n negative examples is significant according to h_{lrs} does not depend on the size of the training set, but only on the class distribution of the examples it covers.

In contrast, h_{chi^2} always fits the same isometric landscape into coverage space. As a result, the evaluation of a point (n, p) depends on its relative location $(n/N, p/P)$. In this case, the same rule will always be evaluated in the same way, as long as it covers the same fraction of the training data. Only the location of the significance cutoff isometric (e.g. 95%) depends on P and N in the case of h_{chi^2}.

7.4. Foil*'s encoding length restriction*

Foil (Quinlan, 1990) uses a criterion based on minimum description length (MDL) for deciding when to stop refining the current rule. For explicitly indicating the p positive examples covered by the rule, one needs h_{MDL} bits:

$$h_{\text{MDL}} = \log_2 (P + N) + \log_2 \binom{P + N}{p}$$

This number is then compared to the number of bits needed to encode the rule, denoted by $l(r)$. If $h_{\text{MDL}}(r) < l(r)$, i.e., if the encoding of the rule is longer than the encoding of the examples themselves, the rule is rejected. As $l(r)$ depends solely on the encoding length (in bits) of the current rule, Foil's stopping criterion—like CN2's significance test—also depends on the size of the training set: the same rule that is too long for a smaller training set might be good enough for a larger training set, in which it covers more examples.

For the purposes of our analysis, we can neglect the rule length as a parameter, and interpret h_{MDL} as a heuristic that is compared to a variable threshold, the size of which depends on the length of the rule. Figure 16 shows the behavior of h_{MDL} in coverage space. The isometric landscape is equivalent to one for the minimum coverage criterion, namely

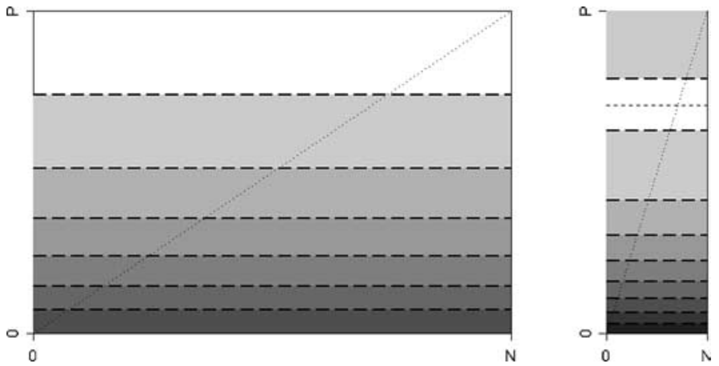

Figure 16. Illustration of Foil's encoding length restriction for domains with $P < N$ (left) and $P > N$ (right). Lighter shades of gray correspond to larger encoding lengths for the rule.

parallel lines to the N-axis. This is not surprising, considering that h_{MDL} is independent of n, the number of covered negative examples.

For $P < N$, h_{MDL} is monotonically increasing with p. As a consequence, for every rule encoding length $l(r)$ there exists a threshold $p(l(r))$ such that for all rules r, $h_{\text{MDL}}(r) > l(r) \Leftrightarrow h_p(r) > p(l(r))$. Thus, in this case, h_{MDL} and h_p are essentially equivalent. The more complicated formulation in the MDL-framework basically serves the purpose of automatically mapping a rule to an appropriate threshold. It remains an open question, whether there is a simpler mapping $r \rightarrow p(r)$ that allows to obtain equivalent (or similar) thresholds without the expensive computation of $l(r)$.

Particularly interesting is the case $P > N$ (right graph of figure 16): the isometric landscape is still the same, but the labels are no longer monotonically increasing. In fact, h_{MDL} has a maximum at the point $p = (P + N)/2$. Below this line (shown dashed in figure 16), the function is monotonically increasing (as above), but above this line it starts to decrease again. Thus, we can formulate the following

Theorem 7.1. *h_{MDL} is compatible with h_p iff $p \leq (P + N)/2$ and it is antagonistic to h_p for $p > (P + N)/2$.*

Proof: (a) $p \leq \frac{P+N}{2}$:

$$\begin{aligned}
h_{\text{MDL}} &= \log_2(P+N) + \log_2\binom{P+N}{p} \\
&\sim \log_2 \frac{\prod_{i=1}^{p}(P+N+1-i)}{\prod_{i=1}^{p} i} \\
&= \sum_{i=1}^{p} \log_2(P+N+1-i) - \sum_{i=1}^{p} \log_2 i \\
&= \sum_{i=1}^{p} (\log_2(P+N+1-i) - \log_2 i)
\end{aligned}$$

The terms inside the sum are all constant. Thus, for two rules r_1 and r_2 with $0 \leq h_p(r_1) < h_p(r_2) \leq \frac{P+N}{2}$, the corresponding sums only differ in the number of terms. As all terms are > 0, $h_p(r_1) < h_p(r_2) \Leftrightarrow h_{\text{MDL}}(r_1) < h_{\text{MDL}}(r_2)$
(b) $p > \frac{P+N}{2}$:

$$h_{\text{MDL}} \sim \log_2 \binom{P+N}{p} = \log_2 \binom{P+N}{P+N-p}$$

Therefore, each rule r with coverage p has the same evaluation as some rule r' with coverage $P+N-p < (P+N)/2$. As the transformation $p \to P+N-p$ is monotonically decreasing,

$$\begin{aligned} h_{\text{MDL}}(r_1) > h_{\text{MDL}}(r_2) \Leftrightarrow h_{\text{MDL}}(r_1') > h_{\text{MDL}}(r_2') \Leftrightarrow \\ \Leftrightarrow h_p(r_1') > h_p(r_2') \Leftrightarrow h_p(r_1) < h_p(r_2) \end{aligned}$$

□

Corollary 7.2. *h_{MDL} is equivalent with h_p iff $P \leq N$.*

Proof: h_{MDL} reaches its maximum at $p = \frac{P+N}{2}$, which is inside coverage space only if $P > N$. □

Thus, for skewed class distributions with many positive examples, there might be cases where a rule r_2 is acceptable, while a rule r_1 that has the same encoding length ($l(r_1) = l(r_2)$), covers the same number or fewer negative examples ($n(r_1) \leq n(r_2)$), but more positive examples ($p(r_1) > p(r_2)$) is *not* acceptable. For example, in the example shown on the right of figure 16, we assumed $P = 48$ and $N = 20$. A rule $r_1 = (0, 48)$ that covers all positive examples and no negative examples would be allowed approximately 62.27 bits, whereas a rule $r_2 = (20, 34)$ that covers only 34 positive examples but all 20 negative examples would be allowed approximately 70.72 bits. The reason is that h_{MDL} is independent of the number of covered negative examples and that $p = 34 = (P + N)/2$ maximizes the second term in h_{MDL}. Thus, if we assume that both rules have the same encoding length (e.g., $l(r_1) = l(r_2) = 65$ bits), r_2 would still be acceptable, whereas the perfect rule r_1 would not. This is very counter-intuitive and sheds some doubts upon the suitability of Foil's encoding length restriction for such domains.

7.5. Fossil's correlation cutoff

Fossil (Fürnkranz, 1994) imposes a threshold upon its correlation metric (Section 5.2). Only rules that evaluate above this threshold are admitted. Figure 17 shows the effect of this threshold for the stopping value 0.3 which appears to perform fairly well over a variety of domains (Fürnkranz, 1997). It can be clearly seen that, like with CN2, the main focus of this approach is to filter out uninteresting rules, i.e., rules whose example

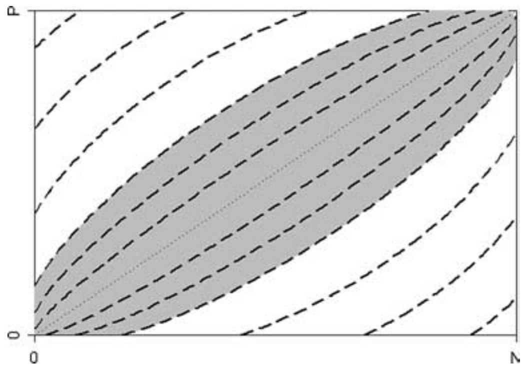

Figure 17. Illustration of Fossil's cutoff criterion.

distribution does not deviate much from the example distribution in the full training set. As discussed in Section 5.2, rules that cover few negative examples and rules that cover many positive examples are preferred, as witnessed by the bended shape of the isometric lines.

Note that Fossil applies this filtering criterion also to *incomplete* rules, i.e., to rules that could still be refined into a better rule. This differs from the approach taken by CN2, where only candidates that appear to be better than the current best rule are tested for significance, but it behaves like Foil, where the encoding length of each candidate rule is tested. Like Foil, Fossil also does not return the rule with the highest evaluation, but it continues to add conditions until the stopping criterion fires. Thus, the cutoff line shown in figure 17 may be viewed as a minimum quality line: learning stops as soon as the path of the learner crosses this line (from the acceptable region to the non-acceptable region), and the last rule above this line is returned.

In this context, it should be noted that the correlation metric as proposed in (Fürnkranz, 1994) is only used for local optimization. The correlation is not computed on the entire training set, but only on the set of examples covered by the predecessor of the rule. As a consequence, Fossil cannot return the rule with the largest correlation (because the correlations for different rules may be computed on different example sets), but it will always return the last rule searched. Thus, as in Foil, the heuristic is only used for determining the best refinement of the currently searched rule, and not for finding an optimum among all candidate rules. For this type of algorithms, the stopping criterion is particularly crucial. Later versions of Fossil adopted a global evaluation, but no strict empirical comparison of the approaches was performed.

8. Discussion and open questions

In this section we discuss various issues that are brought up by the previous analysis, including the relative merits of calculating coverage as an absolute or a relative frequency, overfitting avoidance, and the evaluation of incomplete rules.

8.1. Absolute vs. relative coverage

Evaluation metrics differ in their response to changes in the shape of the coverage space, i.e., to changes in sample size or sample distribution. For example, weighted relative accuracy always assumes cost lines that are parallel to the diagonal of the coverage space. Thus, the relative location of a rule covering a certain number of positive and negative examples may change with changes in the sample distribution, even if the rule's coverage remains the same. It is particularly important to be aware of this property when working with separate-and-conquer algorithms because the covering algorithm constantly modifies the sample distribution by removing the covered examples after each learned rule. Typically, this process will lead to increasingly skewed example distributions because the rules will cover predominantly positive examples, so that their proportion in the remaining examples will decrease.

On the other hand, iso-accuracy lines always have a slope of 1, independent of the actual distribution of negative and positive examples. Thus, the relative costs of rules covering the same number of positive and negative examples will remain the same, irrespective of the sample distribution. This is the property that allows global optimization in nested coverage spaces. Note, however, that a change in the example distribution may also cause a corresponding change in the coverage of the rule. In this case, the location of the rules relative to the isometrics will change with a different sample, even though the isometrics themselves remain in place.

The difference between these two approaches is particularly important when designing stopping criteria: In the first approach (exemplified by Fossil's cutoff criterion) the filtering criterion depends only on the relative coverage of a rule, whereas in the second approach (exemplified by CN2's significance test) it depends on its absolute coverage. Which of the two approaches is preferable depends on whether the number of training examples covered by a rule in the target concept is proportional to the size of the training set or remains approximately the same, independent of the size of the training set. If we assume the existence of an underlying target theory with a fixed number of rules, the former assumption appears to be more logical. However, empirical evidence shows that the number of found rules is typically increasing with the size of the training set (see, e.g., Oates & Jensen, 1998), which could be interpreted as evidence for the latter approach. More work is needed to give more definite answers to these questions.

8.2. Overfitting avoidance

The reader should keep in mind that a learning algorithm typically evaluates a large number of candidate rules, which makes it quite likely that one of them fits the characteristics of the training set by chance. The evaluation of such a rule (and thus its location in coverage space) may change when evaluated on an independent test set. It will be an interesting task for further work to study the differences in the coverage paths of a metric on the training and test sets. Such a study could be helpful in understanding overfitting and possibly in devising heuristics to counter this effect.

In fact, one of the main reasons why the Laplace and m-estimates are favored over precision is because they are less sensitive to noise in the data and more effective in avoiding overfitting (Džeroski & Bratko, 1992). Our interpretation of these estimates as ways of trading off between precision h_{pr} and the linear cost metric h_{costs} supports this view: For large training sets, h_{costs} can be expected to be less prone to overfitting than h_{pr} because it will typically be easy to find a general rule that has a higher evaluation than a rule that fits a single example. For example, there will usually be many rules that have $h_{\text{acc}} = p - n > 1$, while a rule with $h_{\text{pr}} = 1$ will be hard to beat, even if it only covers a single example. However, for small example sets, each rule will only cover a few examples, causing the same type of problems. As small training sets are typically bound to happen at the end of the covering phase, h_{costs} will eventually also overfit.

Another subject for further study is how to equip heuristic functions with a bias against coverage regions that are prone to overfitting. Currently used algorithms handle the overfitting problem exclusively with stopping criteria or with pruning. Our analysis, however, has also shown that criteria such as CN2's significance test or Fossil's correlation cutoff are biased *towards* rules with low coverage. The reason is that these approaches are targeted towards identifying statistically valid deviations from random classification. Our results show that this is not necessarily a good approach for avoiding overfitting.

8.3. Evaluation of incomplete rules

In accordance with most rule learning algorithms, we also tacitly made the assumption that incomplete rules (or incomplete rule sets) should be evaluated in the same way as complete rules (or complete theories). However, it should be noted that this is not necessarily the case: the value of an incomplete rule does not lie in its ability to discriminate between positive and negative examples, but in its potential of being refined into a high-quality rule. For example, Gamberger and Lavrač (2002) argued that for incomplete rules, it is more important to cover many positives (hence a flatter slope is acceptable), while for complete rules it is more important to cover as few negatives as possible (hence a steeper slope). A similar argument has been made by Bradley (1996) who argued that the non-linear isometrics of the χ^2 statistic should be used in order to discriminate classifiers that do "little work" from classifiers that achieve the same accuracy but are preferable in terms of other metrics like sensitivity and specificity.

9. Related work

Rule learning has a long history, but the investigation of search and pruning heuristics has not yet gone beyond its explorative phase. Numerous heuristics have been tried by various authors but independent comparative reviews are very scarce. Fürnkranz (1999) provides a broad survey of heuristics that can be found in the literature but does not attempt to compare them. Lavrač, Cestnik, and Džeroski (1992a, 1992b) offer a limited review for classification rule learning in inductive logic programming, and provide some empirical results and tentative conclusions. A first step towards a theoretical framework for analysis has been made by Lavrač, Flach, and Zupan (1999).

Visualization in ROC-like spaces is primarily used as a tool for finding a suitable set of classifiers under unknown costs (Swets, Dawes, & Monahan, 2000; Provost & Fawcett, 2001), but its importance for analyzing evaluation metrics has been previously recognized as well. We refer to Flach (2003) for a systematic treatment of visualization of evaluation metrics and their isometrics in ROC space.

Bradley (1996) analyzed the χ^2-test on a confusion matrix and concluded that its isometrics are more suitable for comparing classifiers than the isometrics of accuracy. Coverage spaces were—under the name of TP/FP space—already used by Gamberger and Lavrač (2002) for motivating the introduction of an equivalent form of the g-measure (Theorem 4.7). Highly related to our work is the work of Vilalta and Oblinger (2000) who analyzed evaluation metrics by proposing a bias similarity measure based on the area between isometric lines through a fixed point in ROC space, and tried to relate the similarity between metrics to the performance of classifiers that use these metrics. The main difference to our work is that they focused on decision-tree metrics, where the average impurity over all successor nodes is measured, whereas we focus on a rule learning scenario where only the impurity of a single node (the rule) is of interest.

10. Conclusions

In this paper, we analyzed the most common evaluation metrics for separate-and-conquer rule learning algorithms. We looked at heuristics for evaluating rules as well as filtering and stopping criteria, and covered learning of single rules as well as learning of rule sets. Our main tool of analysis, visualization in coverage space (a variant of ROC space) proved to be particularly suitable for understanding both the behavior of heuristic functions and the dynamics of the covering algorithm.

Our results show that there is a surprising number of equivalences and similarities among commonly-used evaluation metrics. For example, we found that the relevant regions of Foil's information gain metric can be reasonably well approximated by a conceptually simpler cost-weighted difference between positive and negative examples, if the precision of the parent clause is used as the cost ratio. In fact, we identified two basic prototype metrics, precision and the above-mentioned linear cost metric, and showed that they follow complementary strategies: precision tries to optimize the area under the ROC curve for unknown misclassification costs, whereas the cost metric tries to directly find the best theory under known costs. We also showed that a straightforward generalization of the well-known m-estimate may be regarded as a means for trading off between these two prototypes.

Stopping and filtering criteria are more diverse and less explored. For example, even though CN2's significance test and Fossil's evaluation metric and stopping criterion use a χ^2-distribution, their isometric landscapes are different and they are not equivalent. In any case, both criteria focus on statistically valid deviations from random guessing (the diagonal of coverage space). While this is certainly a reasonable thing to do in general, our results show that it is questionable whether this is a suitable technique for avoiding overfitting because it does not seem to penalize regions with low rule coverage. On the other hand, simple approaches like thresholding the coverage of a rule, may be quite appropriate for this purpose. In fact, Foil's MDL-based encoding length restriction is equivalent to a variable threshold on the coverage of the positive examples. However, the threshold, which depends on the encoding length of the rule, seems to be problematic for example distributions with predominantly positive examples, where it may prefer a rule with partial coverage over a perfect rule of the same length that covers all positive and no negative examples. Overall, we believe that our analysis has shown that we are still far from a systematic understanding of stopping criteria. The fact that, unlike decision-tree algorithms, most state-of-the-art rule learning algorithms use pruning for noise-handling may not necessarily be a strong indicator for the superiority of this approach, but may also be interpreted as an indicator of the inadequacy of currently used stopping criteria.

In conclusion, we are confident that the analytic technique used in this paper can effectively be turned around for designing new criteria—quite literally—from the drawing board.

Acknowledgments

During this work, Johannes Fürnkranz was supported by an *APART stipend* (no. 10814) of the Austrian Academy of Sciences. We thank the anonymous reviewers for their helpful comments, and Simon Rawles for careful proof-reading.

Notes

1. These algorithms can handle multi-class problems by transforming them into a set of binary learning problems that discriminate each class from the union of all other classes (Clark & Boswell, 1991; Cohen, 1995) or discriminate all pairs of classes (Fürnkranz, 2002).
2. Strictly speaking this would involve the construction of confidence bands as the ROC curve has to be estimated from samples. This is, however, outside the scope of this paper.
3. In previous work (Fürnkranz & Flach, 2003, 2004) we used the term PN-space. However, we prefer the term coverage space as it is more meaningful.
4. An alternative approach is to decrease the weight of covered examples as in Cohen and Singer (1999) and Lavrač et al. (2004).
5. The reader may have noted that for $m \to \infty$, $h_{\text{gm}} \to c$ for all p and n. Thus for $m = \infty$, the function does not have isometrics because all evaluations are constant. However, this is not a problem for the above construction because we are not concerned with the isometrics of the function h_{gm} at the point $m = \infty$, but with the convergence of the isometrics of h_{gm} for $m \to \infty$. In other words, the isometrics of h_{costs} are not equivalent to the isometrics of h_{gm} for $m = \infty$, but they are equivalent to the limits to which the isometrics of h_{gm} converge if $m \to \infty$.
6. This formulation assumes that we are learning in a propositional setting. For relational learning, Foil does not estimate the precision from the number of covered instances, but from the number of *proofs* for those instances.
7. The preferences would even reverse if fractional examples (i.e., coverage counts smaller than 1) are possible. We do not consider this case here.
8. One may say that h_{pr} assumes a different cost model for each point in the space, depending on the relative frequencies of the covered positive and negative examples. Such local changes of cost models are investigated in more detail by Flach (2003).

References

Bradley, A. P. (1996). ROC curves and the χ^2 test. *Pattern Recognition Letters, 17*, 287–294.

Breiman, L., Friedman, J., Olshen, R., & Stone, C. (1984). *Classification and regression trees*. Pacific Grove, CA: Wadsworth & Brooks.

Cendrowska, J. (1987). PRISM: An algorithm for inducing modular rules. *International Journal of Man-Machine Studies, 27:4*, 349–370.

Cestnik, B. (1990). Estimating probabilities: A crucial task in Machine Learning. In L. Aiello (Ed.), *Proceedings of the 9th European Conference on Artificial Intelligence (ECAI-90)* (pp. 147–150). Stockholm, Sweden: Pitman.

Clark, P., & Boswell, R. (1991). Rule induction with CN2: Some recent improvements. In Y. Kodratoff (Ed.), *Proceedings of the 5th European Working Session on Learning (EWSL-91)* (pp. 151–163), Vol. 482 of *Lecture Notes in Artificial Intelligence*. Porto, Portugal: Springer-Verlag.

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

Cohen, W. (1995). Fast effective rule induction. In A. Prieditis & S. Russell (Eds.), *Proceedings of the 12th International Conference on Machine Learning (ML-95)* (pp. 115–123). Lake Tahoe, CA: Morgan Kaufmann.

Cohen, W., & Singer, Y. (1999). A simple, fast, and effective rule learner. In *Proceedings of the 16th National Conference on Artificial Intelligence (AAAI-99)* (pp. 335–342). Orlando, FL: AAAI/MIT Press.

De Raedt, L., & Van Laer, W. (1995). Inductive constraint logic. In K. Jantke, T. Shinohara, & T. Zeugmann (Eds.), *Proceedings of the 5th Workshop on Algorithmic Learning Theory (ALT-95)* (pp. 80–94), Vol. 997 of *Lecture Notes in Artificial Intelligence*. Fukuoka, Japan: Springer-Verlag.

Džeroski, S., & Bratko, I. (1992). Handling noise in Inductive Logic Programming. In S. Muggleton & K. Furukawa (Eds.), *Proceedings of the 2nd International Workshop on Inductive Logic Programming (ILP-92)* (pp. 109–125). Tokyo, Japan.

Ferri, C., Flach, P., & Hernández, J. (2002). Learning decision trees using the area under the ROC curve. In C. Sammut, & A. Hoffmann (Eds.), *Proceedings of the 19th International Conference on Machine Learning (ICML-02)* (pp. 139–146). Sydney, Australia, Morgan Kaufmann.

Flach, P. (2003). The geometry of ROC space: Using ROC isometrics to understand machine learning metrics. In T. Fawcett & N. Mishra (Eds.), *Proceedings of the 20th International Conference on Machine Learning (ICML-03)* (pp. 194–201). Washington, DC: AAAI Press.

Flach, P., & Wu, S. (2003). Repairing concavities in ROC curves. In J. Rossiter & T. Martin (Eds.), *Proceedings of the 2003 UK Workshop on Computational Intelligence* (pp. 38–44). University of Bristol.

Fürnkranz, J. (1994). FOSSIL: A robust relational learner. In F. Bergadano & L. De Raedt (Eds.), *Proceedings of the 7th European Conference on Machine Learning (ECML-94)* (pp. 122–137), Vol. 784 of *Lecture Notes in Artificial Intelligence*. Catania, Italy: Springer-Verlag.

Fürnkranz, J. (1997). Pruning algorithms for rule learning. *Machine Learning, 27:2*, 139–171.

Fürnkranz, J. (1999). Separate-and-conquer rule learning. *Artificial Intelligence Review, 13:1*, 3–54.

Fürnkranz, J. (2002). Round robin classification. *Journal of Machine Learning Research, 2*, 721–747.

Fürnkranz, J., & Flach, P. (2003). An analysis of rule evaluation metrics. In T. Fawcett & N. Mishra (Eds.), *Proceedings of the 20th International Conference on Machine Learning (ICML-03)* (pp. 202–209). Washington, DC: AAAI Press.

Fürnkranz, J., & Flach, P. (2004). An analysis of stopping and filtering criteria for rule learning. In J.-F. Boulicaut, F. Esposito, F. Giannotti, & D. Pedreschi (Eds.), *Proceedings of the 14th European Conference on Machine Learning (ECML-04)*, Vol. 3201 of *Lecture Notes in Artificial Intelligence*. Pisa, Italy: Springer-Verlag.

Fürnkranz, J., & Widmer, G. (1994). Incremental reduced error pruning. In W. Cohen & H. Hirsh (Eds.), *Proceedings of the 11th International Conference on Machine Learning (ML-94)*, (pp. 70–77). New Brunswick, NJ: Morgan Kaufmann.

Gamberger, D., & Lavrač, N. (2002). Expert-guided subgroup discovery: Methodology and application. *Journal of Artificial Intelligence Research, 17*, 501–527.

Giordana, A., & Sale, C. (1992). Learning structured concepts using genetic algorithms. In D. Sleeman & Edwards, P. (Eds.), *Proceedings of the 9th International Workshop on Machine Learning (ML-92)* (pp. 169–178). Edinburgh, United Kingdom, Morgan Kaufmann.

Jovanoski, V., & Lavrač, N. (2001). Classification rule learning with APRIORI-C. In P. Brazdil and A. Jorge (Eds.), *Proceedings of the 10th Portuguese Conference on Artificial Intelligence (EPIA-01)* (pp. 44–51), Vol. 2258 of *Lecture Notes in Artificial Intelligence*. Porto, Portugal: Springer-Verlag.

Kaufman, K., & Michalski, R. (2000). An adjustable rule learner for pattern discovery using the AQ methodology. *Journal of Intelligent Information Systems, 14:2/3*, 199–216.

Klösgen, W. (1996). Explora: A multipattern and multistrategy discovery assistant. In U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, & R. Uthurusamy (Eds.), *Advances in Knowledge Discovery and Data Mining* (pp. 249–271). AAAI Press.

Lavrač, N., Flach, P., & Zupan, B. (1999). Rule evaluation measures: A unifying view. In S. Džeroski & P. Flach (Eds.), *Proceedings of the 9th International Workshop on Inductive Logic Programming (ILP-99)* (pp. 174–185), Vol. 1634 of *Lecture Notes in Artificial Intelligence*. Bled, Slovenia: Springer-Verlag.

Lavrač, N., Kavšek, B., Flach, P., & Todorovski, L. (2004). Subgroup discovery with CN2-SD. *Journal of Machine Learning Research, 5*, 153–188.

Lavrač, N., Cestnik, B., & Džeroski, S. (1992a). Search heuristics in empirical Inductive Logic Programming. In *Logical Approaches to Machine Learning, Workshop Notes of the 10th European Conference on AI*. Vienna, Austria.

Lavrač, N., Cestnik, B., & Džeroski, S. (1992b). Use of heuristics in empirical Inductive Logic Programming. In S. Muggleton & K. Furukawa (Eds.), *Proceedings of the 2nd International Workshop on Inductive Logic Programming (ILP-92)*. Tokyo, Japan.

Liu, B., Hsu, W., & Ma, Y. (1998). Integrating classification and association rule mining. In R. Agrawal, P. Stolorz, & G. Piatetsky-Shapiro (Eds.), *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining (KDD-98)* (pp. 80–86). AAAI Press.

Liu, B., Ma, Y., & Wong, C.-K. (2000). Improving an exhaustive search based rule learner. In D. Zighed, J. Komorowski, & J. Zytkow (Eds.), *Proceedings of the 4th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD-00)*, (pp. 504–509), Vol. 2258 of *Lecture Notes in Artificial Intelligence*. Lyon, France: Springer-Verlag.

Michalski, R. (1969). On the quasi-minimal solution of the covering problem. In *Proceedings of the 5th International Symposium on Information Processing (FCIP-69)* (pp. 125–128), Vol. A3 (Switching Circuits). Bled, Yugoslavia.

Mittenecker, E. (1983). *Planung und statistische Auswertung von Experimenten*, 10th edition. Vienna, Austria: Verlag Franz Deuticke. (In German)

Mladenić, D. (1993). Combinatorial optimization in inductive concept learning. In *Proceedings of the 10th International Conference on Machine Learning (ML-93)*. (pp. 205–211). Amherst, MA: Morgan Kaufmann.

Muggleton, S. (1995). Inverse entailment and Progol. *New Generation Computing, 13:3/4*, 245–286.

Oates, T., & Jensen, D. (1998). Large data sets lead to overly complex models: An explanation and a solution. In R. Agrawal, P. Stolorz, & G. Piatetsky-Shapiro (Eds.), *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining (KDD-98)* (pp. 294–298). AAAI Press.

Pagallo, G., & Haussler, D. (1990). Boolean feature discovery in empirical learning. *Machine Learning, 5:1*, 71–99.

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

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning, 1:1*, 81–106.

Quinlan, J. R. (1990). Learning logical definitions from relations. *Machine Learning, 5:3*, 239–266.

Quinlan, J. R., & Cameron-Jones, R. (1995). Induction of logic programs: FOIL and related systems. *New Generation Computing, 13:3/4*, 287–312.

Swets, J., Dawes, R., & Monahan, J. (2000). Better decisions through science. *Scientific American, 283:4*, 82–87.

Todorovski, L., Flach, P., & Lavrač, N. (2000). Predictive performance of weighted relative accuracy. In D. Zighed, J. Komorowski, & J. Zytkow (Eds.), *Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD-00)* (pp. 255–264), Vol. 2258 of *Lecture Notes in Artificial Intelligence*. Lyon, France: Springer-Verlag.

van Rijsbergen, C. (1979). *Information retrieval*, 2nd edition. London, UK: Butterworths.

Vilalta, R., & Oblinger, D. (2000). A quantification of distance-bias between evaluation metrics in classification. In P. Langley (Ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML-00)*, (pp. 1087–1094). Stanford, CA: Morgan Kaufmann.

Webb, G. (1995). OPUS: An efficient admissible algorithm for unordered search. *Journal of Artificial Intelligence Research, 5*, 431–465.

Weiss, S., & Indurkhya, N. (1991). Reduced complexity rule induction. In J. Mylopoulos, & R. Reiter (Eds.), *Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI-91)* (pp. 678–684). Sydney, Australia: Morgan Kaufmann.

Received November 11, 2003
Revised September 8, 2004
Accepted September 8, 2004
Final manuscript September 8, 2004