An Analysis of Rule Evaluation Metrics

Johannes Fürnkranz JUFFI@OEFAI.AT
Austrian Research Institute for Artificial Intelligence, Schottengasse 3, A-1010 Wien, Austria

Peter A. Flach PETER.FLACH@BRISTOL.AC.UK
Department of Computer Science, University of Bristol, Woodland Road, Bristol BS8 1UB, UK

Abstract

In this paper we analyze the most popular evaluation metrics for separate-and-conquer rule learning algorithms. Our results show that all commonly used heuristics, including accuracy, weighted relative accuracy, entropy, Gini index and information gain, 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.

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 this 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. 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 theories (rule sets) and clauses (rules) in PROLOG.

Various approaches that adhere to this framework differ in the way single rules are learned (Fürnkranz, 1999). The vast majority of algorithms uses a greedy top-down hill-climbing or beam search strategy, other approaches search bottom-up or apply exhaustive or evolutionary search algorithms. Common to all algorithms is that they have to use a metric for evaluating the quality of a candidate rule.

Note that rule learning algorithms that are based on iterative refinement of candidate rules typically use the same metric for evaluating complete and incomplete rules. While the evaluation of complete rules should measure the rule's potential of classifying unseen test cases, the evaluation of an incomplete rule should capture its potential to be refined into a high-quality complete rule. In this case, the evaluation metric is used as a *search heuristic*. We note that, in principle, different types of search heuristics are possible (cf. also Section 5), but, like all refinement-based rule learning algorithms, we will not further differentiate between evaluation metrics and search heuristics, and use the terms interchangeably in the remainder of the paper.

The outline of the paper is as follows. In Section 2 we give some formal definitions used in the rest of the paper. In Section 3 we present our main analysis tool: isometrics in PN-space (a variant of ROC space). Section 4 is the main part of the paper, presenting our analysis of rule learning heuristics through isometric plots. Section 5 discusses the main implications of the analysis, and Section 6 concludes.

2. Formalities

In the remainder of the paper, we use capital letters to denote the total number of positive (P) and negative (N) examples in the training set, whereas $p(r)$ and $n(r)$ are used for the respective number of examples covered by a rule r. Heuristics are two-dimensional functions of the form $h(p, n)$. We use subscripts to the letter h to differentiate between different heuristics. For brevity and readability, we will abridge $h(p(r), n(r))$ with $h(r)$, and omit the argument (r) from functions p, n, and h when it is clear from the context,

Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003), Washington DC, 2003.

Definition 2.1 (compatible) *Two search heuristics 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 2.2 (antagonistic) *Two search heuristics h_1 and h_2 are* antagonistic *iff for all rules r, s:*
$h_1(r) > h_1(s) \Leftrightarrow h_2(r) < h_2(s)$.

Definition 2.3 (equality-preserving) *Two search heuristics 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 2.4 *Compatible or antagonistic search heuristics are equality-preserving.*

Proof: Assume they would not be equality-preserving. This means there exist rules r and s with $h_1(r) = h_2(s)$ 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). This leads to a contradiction. □

Definition 2.5 (equivalence) *Two search heuristics h_1 and h_2 are* equivalent *($h_1 \sim h_2$) if they are either compatible or antagonistic.*

Basically, we consider two heuristics as equivalent if they order a set of candidate rules in the same or the opposite way.

3. PN-spaces and Isometrics

We will visualize the behavior of a search heuristic h by plotting it in a rectangular window with two axes representing the positive and negative examples covered by a rule. In this *PN-space*, a point $(n, p) \in [0, N] \times [0, P]$ represents a rule covering p positive and and n negative examples.[1] With each such point, we associate its heuristic value $h(p, n)$ and draw the isometrics of the function h.

Definition 3.1 (isometric) *An* isometric *of a heuristic h is a line (or curve) in PN-space that connects, for some value c, all points (n, p) for which $h(p, n) = c$.*

The importance of isometrics is reflected in the definitions in the previous section: Equality-preserving search heuristics can be recognized by examining their isometrics and establishing that for each isometric line for h_1 there is an identical isometric line h_2. Compatible (antagonistic) search heuristics can be recognized by investigating corresponding isometrics and establishing that their associated heuristic values are in the same (the opposite) order.

Note that PN-graphs are essentially equivalent to the graphs that are used in ROC analysis (e.g., Provost & Fawcett, 2001): A PN-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 PN-graph are equivalent to its isometrics in ROC-space (Flach, 2003). Nevertheless, PN-graphs have several interesting properties that may be of interest depending on the purpose of the visualization. Table 1 compares some of the properties of PN-curves to those of ROC-curves. A more detailed discussion can be found in (Fürnkranz & Flach, 2003).

Table 1. PN-spaces vs. ROC-spaces.

property	ROC-space	PN-space
x-axis	FPR = $\frac{n}{N}$	n
y-axis	TPR = $\frac{p}{P}$	p
empty theory	$(0, 0)$	$(0, 0)$
correct theory	$(0, 1)$	$(0, P)$
universal theory	$(1, 1)$	(P, N)
resolution	$(\frac{1}{P}, \frac{1}{N})$	$(1, 1)$
slope of diagonal	1	$\frac{P}{N}$
slope of $p = n$ line	$\frac{N}{P}$	1

Of particular interest for the covering approach is the property that PN-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. Therefore, the PN-graph for a subset of a training set can be drawn directly into the PN-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 training examples that are covered by this rule are removed from the training set and the learner calls itself on the remaining examples—can be visualized by a nested sequence of PN-graphs (see Figure 6).

4. Analysis

The ultimate goal of learning is to reach point $(0, P)$ in PN-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. Because of space restrictions, we cannot reference each occurrence in the literature, but we have to refer the reader to the survey (Fürnkranz, 1999) to the longer version of the paper (Fürnkranz & Flach, 2003).

[1] In all figures, we will assume $P < N$. This choice was made for esthetic reasons and does not affect our results.

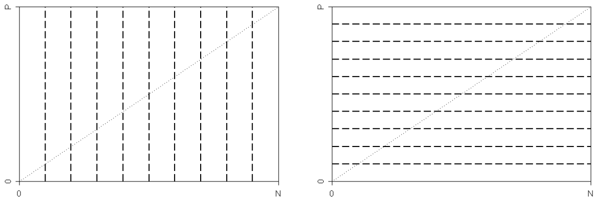

Figure 1. Isometrics for minimizing false positives and for maximizing true positives.

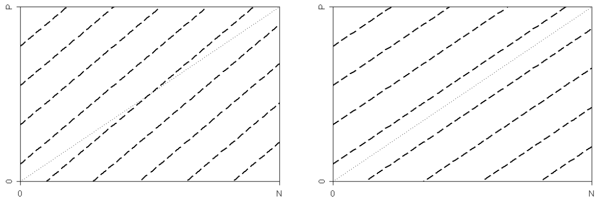

Figure 2. Isometrics for accuracy and weighted relative accuracy

4.1. Basic Heuristics

Clearly, each rule in a correct theory has to cover a subset of the positive examples but none of the negative examples. This property can simply be measured by counting the number of covered negative examples for each individual rule. Alternatively, one can also try to cover all positive examples “at all costs”, i.e., regardless of how many negative examples are covered. This is equivalent to *recall* in information retrieval. Two heuristics that implement these strategies are

$$h_n = -n \qquad h_p = p$$

Figure 1 shows their isometrics: 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.

4.2. Accuracy, WRA, General Costs

Both basic heuristics have the disadvantage that they focus only on one aspect: covering positive examples or excluding negative examples. Ideally, one would like to achieve both goals simultaneously. A straight-forward solution is to simply add up h_n and h_p:

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

The isometrics for this function are shown on the left graph of Figure 2. Note that the isometrics all have a 45^o angle, which means that this heuristic optimizes 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}: $p - n = c_{acc} = c(P + N) - N$. □

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 sample size:

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

The isometrics of this heuristic are shown in the right half of Figure 2. The main difference to accuracy is that the isometrics are now 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 h_{wra} may be viewed as a simplification of *weighted relative accuracy* (Lavrač et al., 1999).

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

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

The two PN-graphs of Figure 2 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

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

Obviously, the accuracy isometrics can be obtained with $a = b = d = 1$ or $c = 1/2$, and the isometrics of weighted relative accuracy can be obtained by setting $a = 1/P$ and $b = 1/N$ or $c = N/(P + N)$ or $d = P/N$. In general, the slope of the parallel isometrics in the PN-graph is $\frac{c-1}{c}$.

4.3. Precision

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

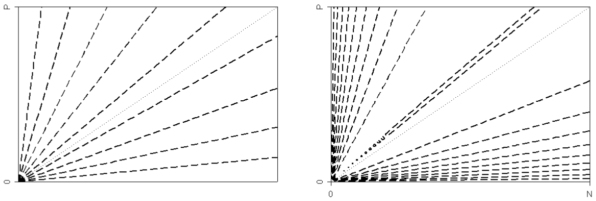

Figure 3. Isometrics for precision and entropy

mining, or *precision* in information retrieval. We will use the latter term:

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

Figure 3 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.

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):

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

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

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

4.4. Information Content, Entropy and Gini index

Some algorithms measure the information content

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

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

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

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 of CN2 (Clark & Niblett, 1989).

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

Entropy is not equivalent to information content and precision, even though it seems to have the same isometrics as these heuristics (see Figure 3). The difference is that the isometrics of entropy go through the undefined point $(0,0)$ and continue on the other side of the 45^o 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 (i.e., it learned decision lists). Thus rules $r = (n,p)$ and $s = (p,n)$ are considered to be of equal quality because if one of them can be used for predicting the positive class, the other can be used 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: $h_{ent} = -h_{pr}\log_2 h_{pr} - (1-h_{pr})\log_2(1-h_{pr})$ with $h_{pr} \in [0,1]$. This function has its maximum at $h_{pr} = 1/2 \Leftrightarrow p = n$. From the fact that it is strictly monotonically increasing for $p \leq n$ follows that $h_{pr}(x) < h_{pr}(y) \Rightarrow h_{ent}(x) < h_{pr}(y)$ in this region. Analogously, $h_{pr}(x) < h_{pr}(y) \Rightarrow h_{ent}(x) > h_{pr}(y)$ for $p \geq n$, where h_{ent} is monotonically decreasing in h_{pr}. □

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

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

The Gini index has the same isometric structure as entropy, it only differs in the distribution of the values (hence the lines of the contour plot are 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} ($h_{gini} = h_{pr}(1-h_{pr})$) and both functions have essentially the same shape. □

4.5. Information Gain

Next, we will look at Foil's version of information gain (Quinlan, 1990), which, 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 the 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

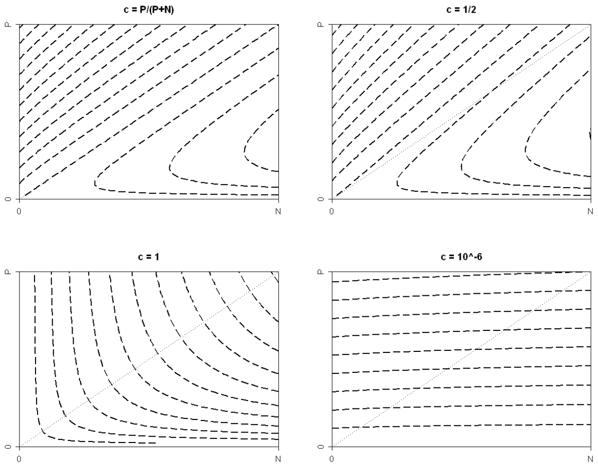

Figure 4. Isometrics for information gain as used in Foil. The curves show different values c for the precision of the parent rule.

examples (as a bias for generality). The exact formula is[2]

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

where $c = h_{pr}(r')$ is the precision of the parent rule. For the following analysis, we will view c as a parameter taking values in the interval $[0, 1]$.

Figure 4 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^o slope, and in the lower two graphs ($c = 1$ and $c = 10^{-6}$) it coincides with the vertical and horizontal axes respectively. From these graphs, it can be seen that above the base line, information gain is equivalent to the linear cost metric h_{costs}.

It is hard to explain the non-linear isometrics below the base line. However, note that this region corresponds to the cases where the precision of the rule is smaller than c, i.e., smaller than the precision of its parent rule. 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 base line has information gain 0, and the linear isometrics above it all have an increasingly positive gain.

These graphs lead us to formulate the following

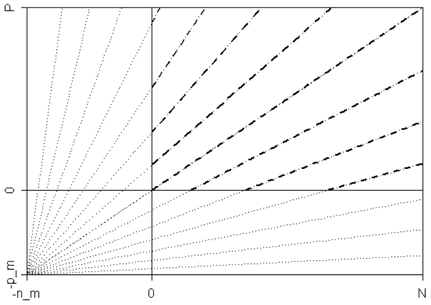

Figure 5. Isometrics for the m-estimate

Conjecture 4.7 *For $p > \frac{c}{1-c}n$: h_{foil} is equivalent to h_{costs} (where $c \in [0, 1]$ is the precision of the parent clause in h_{foil} and $1 - c$ is the cost parameter in h_{costs}).*

4.6. Laplace and m-estimates

The Laplace and m-estimates (Cestnik, 1990) are very common modifications of h_{pr}.

$$h_{lap} = \frac{p+1}{p+n+2} \qquad h_m = \frac{p + m\frac{P}{P+N}}{p+n+m}$$

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 PN-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 5). The resulting pattern of isometrics is symmetric around the line that goes through $(-n_m, -p_m)$ and $(0, 0)$. Thus, the Laplace estimate is symmetric around the 45^o line, while the m-estimate is symmetric around the diagonal of the PN-graph.

Another noticeable effect of the transformation is that the isometrics in the relevant window $(0, 0) - (P, N)$ become increasingly parallel to the symmetry line, the farther the origin moves away from $(0, 0)$. For $m \to \infty$, the isometrics of the m-estimate converge towards the isometrics of relative weighted accuracy (see theorem 4.8 below).

[2] 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.

4.7. 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:

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

The second version of the heuristic basically defines the rotation point by specifying its co-ordinates $(-b, -a)$ in PN-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, $c = 0$ denotes $(0, -m)$, whereas $c = 1$ means $(-m, 0)$. The line that connects the rotation point and $(0, 0)$ has a slope of $\frac{1-c}{c}$. 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}.[3] Therefore, all heuristics discussed in this paper may (at least in their relevant regions) be viewed as equivalent to some instantiation of this general model.

5. Discussion

In the previous section we have identified two fundamental types of rule learning heuristics: precision h_{pr}, which rotates around the origin $(0, 0)$, and h_{costs} which covers the PN-space with parallel lines. We have also seen that h_{gm} can be used for trading off between the two basic models. In this section, we will discuss a few interesting differences between h_{pr} and h_{costs}.

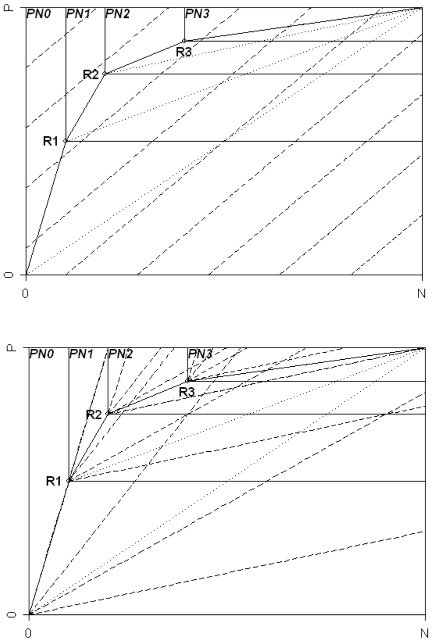

Figure 6. Accuracy and precision in nested PN-spaces.

A property that makes h_{costs} attractive for covering algorithms is that a local optimum in the subspace PN_i, which corresponds to the examples that remain after reaching point R_i, is also optimal in the global PN-space. This is because all isometrics are parallel lines with the same angle, and nested PN-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 $(0, 0)$ of the current subspace PN_i. This is illustrated in Figure 6, where the subspaces PN_i correspond to the situation after removing all examples covered by the rule set $\{R_j | j \leq i\}$.

Also note that at each point (n, p), h_{pr} is equivalent to h_{costs} for $c = \frac{n}{p+n}$ (the slope of the line connecting $(0, 0)$ with (n, p) is $\frac{1-c}{c} = p/n$). Thus, 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.

Why such locally changing costs may nevertheless be a reasonable strategy becomes clear when we look at how successive rules are learned (see Figure 6). h_{pr} needs to be evaluated locally in the PN-space PN_i that results from removing all examples already covered by previously learned

[3] The reader may have noted that for $m \to \infty$, $h_{gm} \to c$ 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 $m \to \infty$.

rules R_i. The metric then picks the rule R_{i+1} that promises the steepest ascent for a continuation of the ROC curve that already leads from the origin to R_i. However, while h_{pr} makes a locally optimal choice for a continuation of the ROC curve, this choice need not be globally optimal because a rule with a slightly worse local evaluation may lead to a much better situation for learning the next rule, and thus eventually to a better overall theory.[4]

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. For example, if the point R_2 in Figure 6 could be reached in one step, h_{acc} would directly go there because it has the better global value, whereas h_{pr} would nevertheless first learn R_1 because it promises a greater area under the ROC curve.

An interesting phenomenon is that several heuristics modify their cost model based on the properties of the PN-space. For example, relative weighted accuracy always assumes costs that are parallel to the main diagonal. Similarly, we have seen that Foil's information gain assumes costs that are parallel to the distribution of examples that are covered by the parent rule of the current rule. In effect, such approaches may be seen as normalizing the example distribution and assuming equal costs for positive and negative misclassification *rates* (as opposed to the misclassifications themselves like accuracy does). As the successive removal of covered examples will necessarily skew the example distribution, this seems to be a particularly good idea for covering approaches. On the other hand, if fewer and fewer positive and negative examples remain, the resolution on the positive axis becomes increasingly problematic, with the limiting case where the true positive rate is either 1 or 0 because there is only one positive example left to cover. It is still largely an open question whether such a normalization is beneficial or not.

We have also ignored the fact that a 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. One of the objectives of a heuristic function should be to counter this phenomenon by giving lower evaluations to rules in regions that can be expected to be particularly sensitive to this overfitting problem. In particular, h_{pr} suffers from overfitting because one can always find a rule that covers a single positive example and no negative example, and such a rule has an optimal value $h_{pr} = 1$. For large example sets, h_{costs} can be expected to be less prone to overfitting because it will typically be easy to find a general rule that has a higher evaluation than a rule that fits a single example (e.g., there will usually be many rules that have $h_{acc} = p - n > 1$). In fact, one of the main reasons why the Laplace and m-estimates are favored over precision was because they are less sensitive to noise. Our interpretation of these estimates as ways of trading off between precision and linear costs supports this view. 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. Typically, the problem of overfitting is addressed with a separate set of heuristics, so-called *stopping criteria*. In principle, stopping criteria decide which point of a ROC-curve should be selected. We plan a separate analysis of this issue in forthcoming work.

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 lies not 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.

Highly related 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.

In addition to the above-mentioned works, we refer to (Flach, 2003) for a systematic treatment of the importance of visualizing evaluation metrics and their isometrics in ROC-space.

[4] A similar idea is used by (Ferri et al., 2002): they suggest to maximize the area under the ROC curve by sorting all rules that correspond to the leaves of a decision tree according to h_{pr}. The main difference is that in their setting the set of rules is fixed, while in the covering approach rules are added incrementally, and thus a different choice for one rule may lead to a completely different theory. However, one could use their method as a post-processor for re-ordering and finding the right subset of the learned rules.

6. Conclusions and Future Work

In this paper, we analyzed the most common search heuristics for classification rule learning algorithms. Our results show that there is a surprising number of equivalences among these metrics. For example, we found that the relevant regions of Foil's information gain metric are equivalent to a conceptually simpler cost-weighted difference between positive and negative examples, where the precision of the parent clause is used as the cost ratio. In fact, we identified two basic prototypes of heuristics, precision and the above-mentioned cost-weighted difference, and showed that they follow complementary strategies: precision tries to optimize the area under 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 straight-forward generalization of the well-known m-estimate may be regarded as a means for trading off between these two prototypes.

We believe that this work contributes to a better understanding of separate-and-conquer rule learning and its heuristics. However, it also raises several questions, which we hope to answer in future work. First, the computation of search heuristics necessarily happens on a training set and is thus prone to overfitting. An ideal search heuristic should correct for such effects. Moreover, we did not pay attention to the aspect that rules and theories are typically grown iteratively, and that the algorithm has to evaluate how likely an incomplete theory or rule can be refined into a complete theory or rule of high quality. Obviously, this is not the same as evaluating the quality of the incomplete rule/theory itself. It might well be that for such tasks, a different type of heuristic is more adequate. In particular, we have found that all heuristics we looked at use a cost model that yields linear cost isometrics (with the exception of Foil's information gain in regions that are not of interest to the learner). It is an open question whether there is a place for search heuristics with non-linear cost isometrics. Finally, we believe that this work facilitates a systematic empirical comparison of search heuristics because one can now focus on comparing properties of the two basic prototypes and study their trade-offs.

Acknowledgments

Part of this work was supported by the EU project *Data Mining and Decision Support for Business Competitiveness: Solomon Virtual Enterprise* (IST-1999-11495). Johannes Fürnkranz is supported by an *APART stipend* (no. 10814) of the Austrian Academy of Sciences. The Austrian Research Institute for Artificial Intelligence is supported by the Austrian Federal Ministry of Education, Science and Culture. Peter Flach was supported by National ICT Australia and the University of New South Wales as a Visiting Research Fellow during completion of this paper. We thank the anonymous reviewers for useful comments and pointers.

References

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

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

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

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

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

Flach, P. A. (2003). The geometry of ROC space: Using ROC isometrics to understand machine learning metrics. *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*.

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

Fürnkranz, J., & Flach, P. (2003). *An analysis of rule learning heuristics* (Technical Report CSTR-03-002). Department of Computer Science, University of Bristol.

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

Lavrač, N., Flach, P., & Zupan, B. (1999). Rule evaluation measures: A unifying view. *Proceedings of the 9th International Workshop on Inductive Logic Programming (ILP-99)* (pp. 174–185). Springer-Verlag.

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

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

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

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