An Empirical Investigation of the Trade-Off Between Consistency and Coverage in Rule Learning Heuristics

Frederik Janssen and Johannes Fürnkranz

TU Darmstadt, Knowledge Engineering Group
Hochschulstraße 10, D-64289 Darmstadt, Germany
[janssen,juffi]@ke.informatik.tu-darmstadt.de

Abstract. In this paper, we argue that search heuristics for inductive rule learning algorithms typically trade off consistency and coverage, and we investigate this trade-off by determining optimal parameter settings for five different parametrized heuristics. This empirical comparison yields several interesting results. Of considerable practical importance are the default values that we establish for these heuristics, and for which we show that they outperform commonly used instantiations of these heuristics. We also gain some theoretical insights. For example, we note that it is important to relate the rule coverage to the class distribution, but that the true positive rate should be weighted more heavily than the false positive rate. We also find that the optimal parameter settings of these heuristics effectively implement quite similar preference criteria.

1 Introduction

Evaluation metrics for rule learning typically, in one way or another, trade off consistency and coverage. On the one hand, rules should be as consistent as possible by only covering a small percentage of negative examples. On the other hand, rules with high coverage tend to be more reliable, even though they might be less precise on the training examples than alternative rules with lower coverage. An increase in coverage of a rule typically goes hand-in-hand with a decrease in consistency, and vice versa. In fact, the conventional top-down hill-climbing search for single rules follows exactly this principle: starting with the empty rule, conditions are greedily added, thereby decreasing coverage but increasing consistency.

In this work, we show that five well-known rule evaluation metrics (a cost trade-off, a relative cost trade-off, the *m*-estimate, the F-measure, and the Klösgen measures) provide parameters that allow to control this trade-off. After a brief discussion of these heuristics, we will report on an extensive experimental study with the goal of determining optimal values for each of their respective parameters, which will allow us to draw some interesting conclusions about heuristic rule learning.

This is the authors' version of the work from www.ke.tu-darmstadt.de. The original publication is available at www.springerlink.com, DOI: 10.1007/978-3-540-88411-8_7, and appeared in Boulicaut, Jean-Franois et al. (Eds.): Proceedings of the 11th International Conference on Discovery Science (DS-08), pp. 40–51, 2008, http://www.springerlink.com/content/5h478u088k513n6w/.

2 Separate-and-Conquer Rule Learning

The goal of an inductive rule learning algorithm is to automatically learn rules that allow to map the examples of the training set to their respective classes. Algorithms differ in the way they learn individual rules, but most of them employ a *separate-and-conquer* or *covering* strategy for combining rules into a rule set [5], including RIPPER [3], arguably one of the most accurate rule learning algorithms today.

Separate-and-conquer rule learning can be divided into two main steps: First, a single rule is learned from the data (the *conquer* step). Then all examples which are covered by the learned rule are removed from the training set (the *separate* step), and the remaining examples are “conquered”. The two steps are iterated until no more positive examples are left. In a simple version of the algorithm this ensures that every positive example is covered at least by one rule (*completeness*) and no negative example is included (*consistency*). More complex versions of the algorithm will allow certain degrees of incompleteness (leaving some examples uncovered) and inconsistencies (covering some negative examples).

For our experiments, we implemented a simple separate-and-conquer rule-learner with a top-down hill-climbing search for individual rules. Rules are greedily refined until no more negative examples are covered, and the best rule encountered in this refinement process (not necessarily the last rule) is returned. We did not employ explicit stopping criteria or pruning techniques for overfitting avoidance, because we wanted to gain a principal understanding of what constitutes a good rule evaluation metric.

3 Rule Learning Heuristics

As discussed above, individual rules should simultaneously optimize two criteria:

Coverage: the number of positive examples that are covered by the rule (p) should be maximized and

Consistency: the number of negative examples that are covered by the rule (n) should be minimized.

Thus, most heuristics depend on p and n, but combine these values in different ways. A few heuristics also include other parameters, such as the length of the rule, but we will not further consider those in this paper. In the following, we will closely follow the terminology and notation introduced in [6]. As an evaluation framework coverage spaces [6], un-normalized ROC spaces, are used in the remainder of this paper. These allow to graphically interpret evaluation metrics by their isometrics.

3.1 Basic Heuristics

true positive rate (recall) $h_{tpr} = h_{Recall} = \frac{p}{P}$

As longer rules typically cover fewer examples, we would argue that this is just another way of measuring coverage. Also, in [7] it was recently found that including rule length does not improve the performance on heuristics that have been derived by meta-learning.

computes the coverage on the positive examples only. It is – on its own – equivalent to simply using p (because P, the total number of positive examples, is constant for a given dataset). Due to its independence of covered negative examples, its isometrics are parallel horizontal lines.

false positive rate $h_{\mathit{fpr}} = \frac{n}{N}$

computes the coverage on the negative examples only (N stands for the total number of negative examples). Its isometrics are parallel vertical lines.

full coverage $h_{\mathit{Coverage}} = \frac{p+n}{P+N}$

computes the fraction of all covered examples. The maximum heuristic value is reached by the universal theory, which covers all examples (the point (N, P) of the coverage space). The isometrics are parallel lines with a slope of -1 (similar to those of the lower right graph in Figure 1).

3.2 Composite Heuristics

The heuristics shown in the previous section only optimize one of the two criteria. Two simple criteria, which try to optimize both criteria are

precision $h_{\mathit{Precision}} = \frac{p}{p+n}$

computes the fraction of correctly classified examples (*p*) among all covered examples (*p*+*n*). Its isometrics rotating around the origin.

weighted relative accuracy (WRA) $h_{\mathit{WRA}} = h_{\mathit{tpr}} - h_{\mathit{fpr}}$

computes the difference between the true positive rate and the false positive rate. The upper middle graph of Figure 1 shows the isometrics of WRA.

However, these two heuristics are known to have complementary disadvantages. Precision is known to overfit the data, i.e., to strongly prefer consistency over coverage. Conversely, the experimental evidence given in [11], which is consistent with our own experience, suggests that WRA has a tendency to overgeneralize, i.e., that it places too strong emphasis on coverage.

Thus, it is necessary to find the right trade-off between consistency and coverage. Many other heuristics implement fixed trade-offs between these criteria. In the next section, we will discuss five heuristics that allow to tune this trade-off with a parameter.

3.3 Parametrized Heuristics

In this section we show that the heuristics which we consider in this work all have a parameter that trades off consistency for coverage, but do so in different forms. The two cost measures directly trade off absolute or relative positive and negative coverage. Thereafter, we will see three measures that use $h_{\mathit{Precision}}$ for optimizing consistency, but use different measures ($h_{\mathit{Recall}}, h_{\mathit{WRA}}, h_{\mathit{Coverage}}$) for optimizing coverage.

cost measure $h_{\mathit{cost}} = c \cdot p - (1-c) \cdot n$

allows to directly trade off consistency and coverage with a parameter $c \in [0, 1]$. $c = 0$ only considers consistency, $c = 1$ only coverage. If $c = 1/2$, the resulting heuristic ($h_{\mathit{Accuracy}} = p - n$) is equivalent to **accuracy**, which computes the percentage of correctly classified examples among all training examples. The isometrics of this heuristics are parallel lines, with a slope of ${}^{(1-c)}/_{c}$.

relative cost measure $h_{\mathit{rcost}} = c_r \cdot h_{\mathit{tpr}} - (1-c_r) \cdot h_{\mathit{fpr}}$

trades off the true positive rate and the false positive rate. This heuristic is quite similar to h_{cost}. In fact, for any particular data set, one can choose $c = \frac{N}{P+N} \cdot c_r$ to transform the cost measure into the relative cost measure. However, this normalization may (and will) make a difference if the same value is used across a wide variety of datasets with different class distributions. Clearly, setting $c_r = 1/2$ is compatible (as defined in [6]) with WRA.

F-measure $h_{\mathit{F\text{-}Measure}} = \frac{(\beta^2+1) \cdot h_{\mathit{Precision}} \cdot h_{\mathit{Recall}}}{\beta^2 \cdot h_{\mathit{Precision}} + h_{\mathit{Recall}}}$

The F-measure [10] has its origin in Information Retrieval and trades off the basic heuristics $h_{\mathit{Precision}}$ and h_{Recall}. Basically, the isometrics (for an illustration see [6]) are identical to those of precision, with the difference that the rotation point is not in the point $(0,0)$ but in a point $(-g,0)$, where g depends on the choice of β. If $\beta \to 0$, the origin moves towards $(0,0)$, and the isometrics correspond to those of $h_{\mathit{Precision}}$. The more the parameter is increased the more the origin of the isometrics is shifted in the direction of the negative N-axis. The observable effect is that the lines in the isometrics becomes flatter and flatter. Conversely if $\beta \to \infty$ the resulting isometrics approach those of h_{Recall} which are horizontal parallel lines.

m-estimate $h_{\mathit{m\text{-}estimate}} = \frac{p+m \cdot \frac{P}{P+N}}{p+n+m}$

The idea of this parametrized heuristic [2] is to presume that a rule covers m training examples *a priori*, maintaining the distribution of the examples in the training set ($m \cdot P/(P+N)$ examples are positive). For $m = 2$ and assuming an equal example distribution ($P = N$), we get the **Laplace** heuristic h_{Laplace} as a special case.

If we inspect the isometrics in relation to the different parameter settings, we observe a similar behavior as discussed above for the F-measure, except that the origin of the turning point now does not move on the N-axis, but it is shifted in the direction of the negative diagonal of the coverage space (cf. [6] for an illustration). $m = 0$ corresponds to precision, and for $m \to \infty$ the isometrics become increasingly parallel to the diagonal of the coverage space, i.e., they approach the isometrics of h_{WRA}. Thus, the m-estimate trades off $h_{\mathit{Precision}}$ and h_{WRA}.

Klösgen $h_{\mathit{Klösgen}} = (h_{\mathit{Coverage}})^{\omega} \cdot \left(h_{\mathit{Precision}} - \frac{P}{P+N}\right)$

This family of measures was first proposed in [9] and trades off *Precision Gain* (the increase in precision compared to the default distribution $P/(P+N)$) and *Coverage*. The isometrics of *Precision Gain* on its own behave like the isometrics of precision, except that their labels differ (the diagonal now always corresponds to a value of 0).

Setting $\omega = 1$ results in WRA, and $\omega = 0$ yields *Precision Gain*. Thus, the Klösgen measure starts with the isometrics of $h_{\mathit{Precision}}$ and first evolves into those of h_{WRA}, just like the m-estimate. However, the transformation takes a different route, with non-linear isometrics. The first two graphs of Figure 1 show the result for the parameter settings $\omega = 0.5$ and $\omega = 1$ (WRA), which were suggested by Klösgen [9].

With a further increase of the parameter, the isometrics converge to h_{Coverage}. The middle left graph shows the parameter setting $\omega = 2$, which was suggested in [13]. Contrary to the previous settings, the isometrics now avoid regions of low coverage, because low coverage is more severely penalized. A further increase of the parameter results in sharper and sharper bends of the isometrics. The influence of WRA (the part

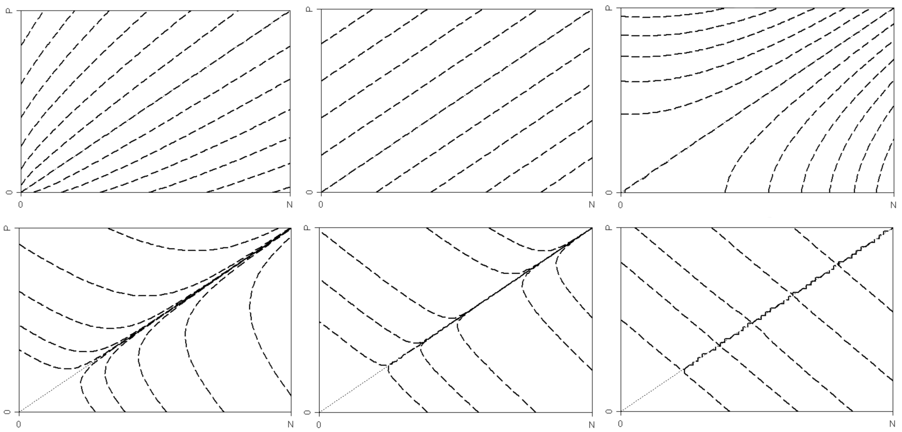


Fig. 1. Klösgen-Measure for $\omega = 0.5, 1, 2, 7, 30, 500$

parallel to the diagonal) vanishes except for very narrow regions around the diagonal, and the isometrics gradually transform into those of coverage.

Another interesting variation of the Klösgen measure is to divide $h_{Coverage}$ by $1 - h_{Coverage}$ instead of raising it to the ω-th power. It has been shown before [9] that this is equivalent to **correlation** ($h_{Corr} = \frac{p\cdot(N-n)-n\cdot(P-p)}{\sqrt{P\cdot N\cdot(p+n)\cdot(P-p+N-n)}}$).

4 Experimental setup

The primary goal of our experimental work was to determine settings for the parametrized heuristics that are optimal in the sense that they will result in the best classification accuracy on a wide variety of datasets. Clearly, the optimal setting for individual datasets may vary.

We arbitrarily selected 27 *tuning datasets* from the UCI-Repository [1] for determining the optimal parameters. To check the validity of the found parameter settings, we selected 30 additional *validation datasets*. The names of all 57 datasets could be found in [8].

The performance on individual datasets was evaluated with a *10-fold stratified Cross Validation* implemented in *Weka* [12]. As we have a large number of different individual results, a key issue is how to combine them into an overall value. We have experimented with several choices. Our primary method was the *macro-averaged accuracy* of one parametrization of a parametrized heuristic which is defined by the sum of all accuracies (the fraction of correctly classified examples among all examples) of the datasets normalized with the number of datasets. This method gives the same weight to all datasets. Alternatively, one could also give the same weight to each example, which results in *micro-averaged accuracy*. It is defined as the sum of all correctly classified examples divided by the total number of examples among all datasets. In effect, this

Algorithm 1 SEARCHBESTPARAMETER($a, b, i, h, dataSets$)

```
$acc_{former} = acc_{best}$                                   # global params
$params$ = CREATELIST($a, b, i$)                              # initialize candidate params
$p_{best}$ = GETBESTPARAM($h, params, dataSets$)
$acc_{best}$ = GETACCURACY($p_{best}$)
# stop if no substantial improvement ($t = 0.001$)
if $acc_{best} - acc_{former} < t$ then
    return ($p_{best}$)
end if
# continue the search with a finer resolution
SEARCHBESTPARAMETER($p_{best} - \frac{i}{2}, p_{best} + \frac{i}{2}, \frac{i}{10}, h, dataSets$)
```

method assigns a higher weight to datasets with many examples, whereas those with few examples get a smaller weight.

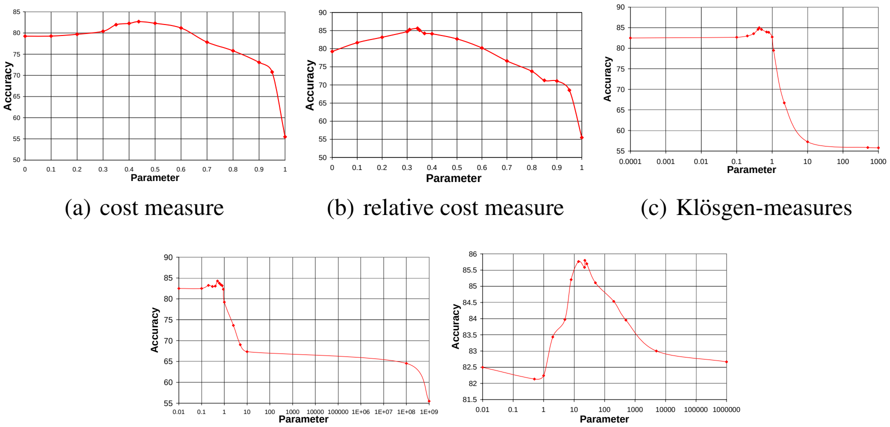
As there are large differences in the variances of the accuracies of the individual datasets, one could also focus only on the *ranking* of the heuristics and neglect the magnitude of the accuracies on different datasets. Small random variations in ranking performance will cancel out over multiple datasets, but if there is a consistent small advantage of one heuristic over the other this will be reflected in a substantial difference in the average rank (the sum of individual ranks normalized by the number of datasets). Finally, we also measured the *size* of the learned theories by the average number of conditions.

5 The Search Strategy

This section describes our method for searching for the optimal parameter setting. Our expectation was that for all heuristics, a plot of accuracy over the parameter value will result in an inverse U-shape, i.e., there will be overfitting for small parameter values and over-generalization for large parameter values, with a region of optimality inbetween. Thus, we adopted a greedy search algorithm that continuously narrows down the region of interest. First, it tests a wide range of intuitively appealing parameter settings to get an idea of the general behavior of each of the five parametrized heuristics. The promising parameters were further narrowed down until we had a single point that represents a region of optimal performance.

Algorithm 1 shows the search procedure in detail. We start with a lower (a) and upper (b) bound of the region of interest, and sample the space between them with a certain interval width i. For measures with parameter space $[0, \infty)$ we used a logarithmic scale. For each sampled parameter value, we estimate its macro-averaged accuracy on all tuning datasets, and, based on the obtained results, narrow down the values a, b, and i.

Intuitively, the farther the lower border a and the upper border b of the interval are away from the best parameter p_{best}, and the denser the increment, the better are our chances to find the optimal parameter, but the higher are the computational demands. As a compromise, we used the following approach for adjusting the values of these



(a) cost measure (b) relative cost measure (c) Klösgen-measures

(d) F-measure (e) m-estimate

Fig. 2. Macro-averaged Accuracy over parameter values for the five parametrized heuristics

parameters:

$$a \leftarrow p_{best} - \frac{i}{2},\ b \leftarrow p_{best} + \frac{i}{2} \ \text{ and } \ i \leftarrow \frac{i}{10}$$

This procedure is repeated until the accuracy does not increase significantly. As we compare macro-averaged accuracy values over several datasets, we adopted a simple approach that stops whenever the accuracy improvement falls below a threshold $t = 0.001$.

Obviously, the procedure is greedy and not guaranteed to find a global optimum. In particular, there is a risk to miss the best parameter due to the fact that the global best parameter may lie under or above the borders (if the best one so far is 1 for example, the interval that would be searched is $[0.5, 1.5]$; if the global optimum is 0.4, it would not be detected). Furthermore, we may miss a global optimum if it hides between two apparently lower values. If the curve is smooth, these assumptions are justified, but on real-world data we should not count on this. The second point can be addressed by keeping a list of candidate parameters that are all refined and from which the best one is selected. Hence it has to be defined how many candidates should be maintained. Therefore it is necessary to introduce a threshold that discriminates between a normal and a candidate parameter. It is not trivial to determine such a threshold. Due to this the number of candidate parameters is limited to 3 (all experiments confirmed that this is sufficient). The first problem could be addressed by re-searching the entire interval at a finer resolution, but, for the sake of efficiency, we chose the more efficient version.

However, also note that it is not really important to find an absolute global optimum. If we can identify a region that is likely to contain the best parameter for a wide variety of datasets, this would already be sufficient for our purposes. We interpret the found values as good representatives for optimal regions.

6 Results

In this section we focus on the results of the search for optimal parameter values. We will illustrate the average accuracy of the different heuristics under various parameter settings, identify optimal parameters, compare their isometrics, and evaluate their general validity.

6.1 Optimal parameters for the five heuristics

Our first goal was to obtain optimal parameter settings for the five heuristics. As discussed above, the found values are not meant to be interpreted as global optima, but as representatives for regions of optimal performance. Figure 2 shows the obtained performance curves.

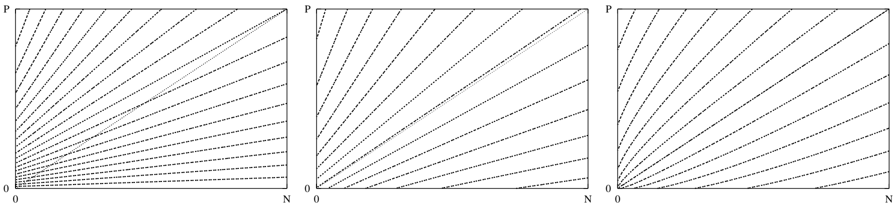
Cost Measures Figures 2 (a) and (b) show the results for the two cost measures. Compared to the other measures, these curves are comparably smooth, and optimal values could be identified quite easily. Optimizing only the consistency (i.e., minimizing the number of negative examples without paying attention to the number of covered positives) has a performance of close to $80\,\%$. Not surprisingly, this can be improved considerably for increasing values of the parameters c and c_r. The best performing values were found at $c = 0.437$ (for the cost metric) and $c_r = 0.342$ (for the relative cost metric). Further increasing these values will decrease performance because of over-generalization. If the parameter approaches 1, there is a steep descent because optimizing only the number of covered examples without regard to the covered negatives is, on its own, a very bad strategy.

It is interesting to interpret the found values. Note, for example, that weighted relative accuracy, which has been previously advocated as rule learning heuristic [11], corresponds to a value of $c_r = 0.5$, equally weighting false positive rate and true positives rate. Comparing this to the optimal region for this parameter, which is approximately between 0.3 and 0.35, it can be clearly seen that it pays off to give a higher weight to the true positive rate.

This is confirmed by the results on the cost metric. The optimal value $c = 0.437$ corresponds to a ratio of positive to negative examples of $\nicefrac{P}{N} = \nicefrac{1-c}{c} \approx 1.29$. In reality, however, for most example sets $P < N$ (for multi-class datasets we assume that P is the number of examples in the largest class). Thus, positive examples have to be given a higher weight than negative examples.

It is also interesting to compare the results of the absolute and relative cost measures: although, as we have stated above, the two are equivalent in the sense that for each individual dataset, one can be transformed into each other by picking an appropriate cost factor, the relative cost measure has a clearly better peak performance exceeding 85%. Thus, it seems to be quite important to incorporate the class distribution $\nicefrac{P}{(P+N)}$ into the evaluation metric. This is also confirmed by the results of $h_{\textit{m-estimate}}$ and $h_{\textit{Klösgen}}$.

Interestingly, the optimal value of $c = 0.342$ corresponds almost exactly to the micro-averaged default accuracy of the largest class (for both tuning and validation datasets). We are still investigating whether this is coincidental or not.



(a) F-measure (b) m-Estimate (c) Klösgen

Fig. 3. Isometrics of the best parameter settings

Klösgen measures Figure 2 (c) shows the results for the Klösgen measures. In the region from 0.1 to 0.4 the accuracy increases continuously until it reaches a global optimum at 0.4323, which achieves an average accuracy of almost $85\,\%$. After the second iteration of the SearchBestParameter algorithm, no better candidate parameters than 0.4 were found. The accuracy decreases again with parametrizations greater than 0.6. As illustrated in Figure 1, the interval $[0, 1]$ describes the trade-off between *Precision* ($\omega = 0$) and WRA ($\omega = 1$), whereas values of $\omega > 1$ trade off between WRA and *Coverage*. The bad performance in this region (presumably due to over-generalization) surprised us, because we originally expected that the behavior that is exhibited by the Klösgen measure for $\omega = 2$, namely to avoid low coverage regions, is preferable over the version with $\omega = 0.5$, which has a slight preference for these regions (cf. Figure 1).

***F*-measure** For the F-measure the same interval as with the Klösgen measures is of special interest (Figure 2 (d)). Already after the first iteration, the parameter 0.5 turned out to have the highest accuracy of $82.2904\,\%$. A better one could not be found during the following iterations. After the second pass two other candidate parameters, namely 0.493 with $84.1025\,\%$ and 0.509 with $84.2606\,\%$ were found. But both of them could not be refined to achieve a higher accuracy and were therefore ignored. The main difference between the Klösgen measures and the F-measure is that for the latter, the accuracy has a steep descent at a very high parametrization of $1 \cdot E^9$. At this point it overgeneralizes in the same way as the Klösgen measures or the cost measures.

***m*-estimate** The behavior of the m-estimate differs from the other parametrized heuristics in several ways. In particular, it proved to be more difficult to search. For example, we can observe a small descent for low parameter settings (Figure 2 (e)). The main problem was that the first iteration exhibited no clear tendencies, so the region in which the best parameter should be could not be restricted.

As a consequence, we re-searched the interval $[0, 35]$ with a smaller increment of 1 because all parameters greater than 35 got accuracies under $85.3\,\%$ and we had to restrict the area of interest. After this second iteration there were 3 candidate parameters, from which 14 achieves the greatest accuracy. After a second run, 23.5 became optimal, which illustrates that it was necessary to maintain a list of candidate parameters. After a

Table 1. Comparison of various results of the optimal parameter settings of the five heuristics (identified by their parameters), other commonly used rule learning heuristics, and JRip (Ripper) with and without pruning, sorted by their macro-averaged accuracy.

(a) on the 27 tuning datasets

	average accuracy		average	
Heuristic	Macro	Micro	Rank	Size
$m = 22.466$	85.87	93.87 (1)	4.54 (1)	36.85 (4)
$c_r = 0.342$	85.61	92.50 (6)	5.54 (4)	26.11 (3)
$\omega = 0.4323$	84.82	93.62 (3)	5.28 (3)	48.26 (8)
JRip	84.45	93.80 (2)	5.12 (2)	16.93 (2)
$\beta = 0.5$	84.14	92.94 (5)	5.72 (5)	41.78 (6)
JRip-P	83.88	93.55 (4)	6.28 (6)	45.52 (7)
Correlation	83.68	92.39 (7)	7.17 (7)	37.48 (5)
WRA	82.87	90.43 (12)	7.80 (10)	14.22 (1)
$c = 0.437$	82.60	91.09 (11)	7.30 (8)	106.30 (12)
Precision	82.36	92.21 (9)	7.80 (10)	101.63 (11)
Laplace	82.28	92.26 (8)	7.31 (9)	91.81 (10)
Accuracy	82.24	91.31 (10)	8.11 (12)	85.93 (9)

(b) on the 30 validation datasets

	average accuracy		average	
Heuristic	Macro	Micro	Rank	Size
JRip	78.98	82.42 (1)	4.72 (1)	12.20 (2)
$c_r = 0.342$	78.87	81.80 (3)	5.28 (3)	25.30 (3)
$m = 22.466$	78.67	81.72 (4)	4.88 (2)	46.33 (4)
JRip-P	78.50	82.04 (2)	5.38 (4)	49.80 (6)
$\omega = 0.4323$	78.46	81.33 (6)	5.67 (6)	61.83 (8)
$\beta = 0.5$	78.12	81.52 (5)	5.43 (5)	51.57 (7)
Correlation	77.55	80.91 (7)	7.23 (8)	47.33 (5)
Laplace	76.87	79.76 (8)	7.08 (7)	117.00 (10)
Precision	76.22	79.53 (9)	7.83 (10)	128.37 (12)
$c = 0.437$	76.11	78.93 (11)	8.15 (11)	122.87 (11)
WRA	75.82	79.35 (10)	7.82 (9)	12.00 (1)
Accuracy	75.65	78.47 (12)	8.52 (12)	99.13 (9)

few more iterations, we found the optimal parameter at 22.466. The achieved accuracy of $85.87\,\%$ was the optimum among all heuristics.

6.2 Behavior of the optimal heuristics

In this section, we compare the parameters which have been found for the five heuristics (cf. also Table 1). In terms of macro-averaged accuracy, the m-estimate and the relative cost measure clearly outperformed the other parametrized heuristics, as well as a few standard heuristics, which we had also briefly mentioned in section 3.3). Interestingly, the relative cost measure performs much worse with respect to micro-averaged accuracy, indicating that it performs rather well on small datasets, but worse on larger datasets. These two heuristics also outperform JRIP (the WEKA-implementation of RIPPER [3]) on the tuning datasets, but, as we will see further below, this performance gain does not quite carry over to new, independent datasets.

Figure 3 shows the isometrics of the best parameter settings of the m-estimate, the F-measure, and the Klösgen-measure. Interestingly, we can see that—within the confinements of their different functionals—all measures try to implement a very similar heuristic. Minor differences are detectable in the low coverage region, where the F-measure is necessarily parallel to the N-axis and the isometrics of the Klösgen measures are slightly bended.

6.3 Validity of the results

In order to make sure that our results are not only due to overfitting of the 27 tuning datasets, we also evaluated the found parameter values on 30 new validation datasets.

Because of space limitations, we omit the corresponding figures for the cost metrics, but they are just parallel lines with slopes that are determined by their respective optimal parameter values (and, in the case of the relative cost measure, also by the class distribution).

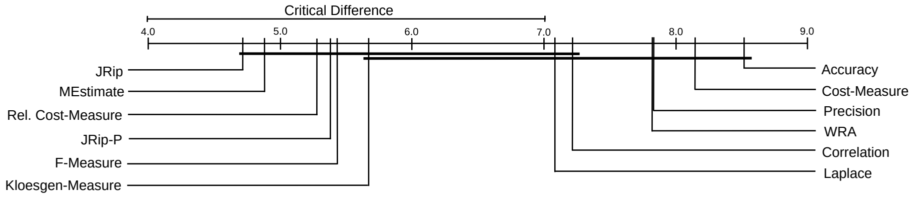


Fig. 4. Comparison of all classifiers against each other with the Nemenyi test. Groups of classifiers that are not significantly different (at $p = 0.05$) are connected.

The results are summarized in Table 1 for both the tuning datasets (left) and the test datasets (right). The numbers in brackets describe the rank of each heuristic according to the measure of the respective column.

Qualitatively, we can see that the relative performance of the heuristics in comparison to each other, and in comparison to the standard heuristics does not change much, with the exception of the considerably better performance of JRIP, which indicates that some amount of overfitting has happened in the optimization phase. However, the performance of the best metrics is still comparable to the performance of JRIP, although the latter achieves this performance with much smaller rule sizes.

Figure 4 displays a comparison of all classifiers done with the Nemenyi test suggested in [4]. All tuned heuristics (except the cost measure) outperform the standard heuristics which is indicated by the large gap between them. The Klösgen measure is the only parametrized heuristic which is not significantly better than the Accuracy heuristic.

7 Conclusions

The experimental study reported in this paper has provided several important insights into the behavior of greedy inductive rule learning algorithms. First, we have determined suitable default values for commonly used parametrized evaluation metrics such as the m-estimate. This is of considerable practical importance, as we showed that these new values outperformed conventional search heuristics and performed comparably to the RIPPER rule learning algorithm. Second, we found that heuristics which take the class distribution into account (e.g., by evaluate relative coverage instead of absolute coverage) outperform heuristics that ignore the class distribution (e.g., the F-measure which trades off recall and precision). Third, however, we found that for a good overall performance, it is necessary to weight the true positive rate more heavily than the false positive rate. This is most obvious in the optimal parameter value for the relative cost metric, but can also be observed in other well-performing heuristics, whose isometrics have a very steep slope in the important regions. Last but not least, we think that this has been the most exhaustive experimental comparison of different rule learning heuristics to date, yielding new insights into their comparative performance.

However, our results also have their limitations. For example, we have only evaluated overall performance over a wide variety of datasets. Obviously, we can expect a better performance if the parameter values are tuned to each individual dataset. We think that the good performance of RIPPER is due to the flexibility of post-pruning, which allows to adjust the level of generality of a rule to the characteristic of a particular dataset. We have deliberately ignored the possibility of pruning for this set of experiments, because our goal was to gain a principal understanding of what constitutes a good rule evaluation metric for separate-and-conquer learning. It is quite reasonable to expect that pruning strategies could further improve this performance. In particular, it can be expected that the performance of parameter values that result in slight overfitting can be considerably improved by pruning (whereas pruning can clearly not help in the case of over-generalization). We are currently investigating this issue.

Acknowledgements

This research was supported by the *German Science Foundation (DFG)* under grant FU 580/2-1.

References

[1] A. Asuncion and D. Newman. UCI machine learning repository, 2007. URL http://www.ics.uci.edu/∼mlearn/MLRepository.html.

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

[3] W. W. Cohen. Fast Effective Rule Induction. In A. Prieditis and S. Russell, editors, *Proceedings of the 12th International Conference on Machine Learning*, pages 115–123, Tahoe City, CA, July 9–12, 1995. Morgan Kaufmann. ISBN 1-55860-377-8. URL http://citeseer.nj.nec.com/cohen95fast.html.

[4] J. Demsar. Statistical comparisons of classifiers over multiple datasets. *Machine Learning Research*, (7):1–30, 2006.

[5] J. Fürnkranz. Separate-and-Conquer Rule Learning. *Artificial Intelligence Review*, 13(1): 3–54, February 1999. URL citeseer.ist.psu.edu/26490.html.

[6] J. Fürnkranz and P. A. Flach. ROC 'n' Rule Learning - Towards a Better Understanding of Covering Algorithms. *Machine Learning*, 58(1):39–77, January 2005. ISSN 0885-6125. URL http://www.cs.bris.ac.uk/Publications/Papers/2000264.pdf.

[7] F. Janssen and J. Fürnkranz. On meta-learning rule learning heuristics. In *Proceedings of the 7th IEEE Conference on Data Mining (ICDM-07)*, pages 529–534, Omaha, NE, 2007.

[8] F. Janssen and J. Fürnkranz. An empirical quest for optimal rule learning heuristics. Technical Report TUD-KE-2008-01, Knowledge Engineering Group, TU Darmstadt, 2008. URL http://www.ke.informatik.tu-darmstadt.de/publications/reports/tud-ke-2008-01.pdf.

[9] W. Klösgen. Problems for Knowledge Discovery in Databases and their Treatment in the Statistics Interpreter Explora. *International Journal of Intelligent Systems*, 7:649–673, 1992.

[10] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA, 1986. ISBN 0070544840.

[11] L. Todorovski, P. Flach, and N. Lavrac. Predictive performance of weighted relative accuracy. In D. A. Zighed, J. Komorowski, and J. Zytkow, editors, *4th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD2000)*, pages 255–264. Springer-Verlag, September 2000. ISBN 3-540-41066-X. URL http://www.cs.bris.ac.uk/Publications/Papers/1000516.pdf.

[12] I. H. Witten and E. Frank. *Data Mining — Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers, 2nd edition, 2005. URL http://www.cs.waikato.ac.nz/ ml/weka/.

[13] S. Wrobel. An Algorithm for Multi-relational discovery of Subgroups. In J. Komorowski and J. Zytkow, editors, *Proc. First European Symposion on Principles of Data Mining and Knowledge Discovery (PKDD-97)*, pages 78–87, Berlin, 1997. Springer Verlag.