From Local to Global Patterns: Evaluation Issues in Rule Learning Algorithms

Johannes Fürnkranz

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

Abstract. Separate-and-conquer or covering rule learning algorithms may be viewed as a technique for using local pattern discovery for generating a global theory. Local patterns are learned one at a time, and each pattern is evaluated in a local context, with respect to the number of positive and negative examples that it covers. Global context is provided by removing the examples that are covered by previous patterns before learning a new rule. In this paper, we discuss several research issues that arise in this context. We start with a brief discussion of covering algorithms, their problems, and review a few suggestions for resolving them. We then discuss the suitability of a well-known family of evaluation metrics, and analyze how they trade off coverage and precision of a rule. Our conclusion is that in many applications, coverage is only needed for establishing statistical significance, and precision is the metric that should be optimized for rules. The main problem with optimizing precision is its unreliability for low example sizes, which is mainly caused by overfitting. We then report some preliminary experiments that addresses this problem by meta-learning a predictor for the true accuracy of a rule based on its coverage on the training set.

1 Introduction

Numerous evaluation heuristics have been proposed for evaluating rules in the context of classification rule learning, subgroup discovery and association rule discovery (Fürnkranz, 1999; Lavrač et al., 1999; Tan et al., 2002). Nevertheless, the issue is not yet well understood. The similarities and differences between the different measures are not explored in sufficient depth, and it is often also not clear, what properties we want an evaluation measure to have. Our research aims at increasing our understanding of these issues through theoretical analysis as well as empirical evaluation.

In this paper, we first discuss the relation between classification rule discovery using the separate-and-conquer or covering strategy and the local pattern discovery task (Section 2). Following up on (Fürnkranz and Flach, 2005), our main tool for analysis will be visualization in coverage space (Section 3). In the following (Section 4), we will analyze a family of well-known rule evaluation measures that have been proposed for subgroup discovery (Klösgen, 1992; Wrobel, 1997), that all have in common that they trade off coverage and precision of the rule, but differ in the weight that they allot to each component. We will argue that, with increasing coverage, the relative importance of coverage becomes negligible. One of the main reasons for including coverage is the

```
function COVERING(Examples)

# initialize the classifier
GlobalClassifier $\leftarrow \emptyset$

# loop until all examples are covered
while Examples $\neq \emptyset$

    # find the best local pattern
    LocalPattern $\leftarrow$ FINDBESTLOCALPATTERN(Examples)

    # add the local pattern to the classifier
    GlobalClassifier $\leftarrow$ GlobalClassifier $\cup$ LocalPattern

    # remove the covered examples
    Examples $\leftarrow$ Examples $\setminus$ COVERED(LocalPattern,Examples)

return GlobalClassifier
```

Fig. 1. The Covering Algorithm

problem of overfitting, to which precision is particularly susceptible. In the final part of the paper (Section 5), we address this problem in a novel way, namely by using training set statistics of a rule for predicting its test performance.

2 From Local to Global Patterns

The covering or separate-and-conquer strategy for inductive rule learning—see (Fürnkranz, 1999) for a survey—may be viewed as a general approach for combining local patterns into global classifiers. The basic idea is to repeatedly find the best local pattern and add it to a growing theory. The goodness of the local pattern is measured with some heuristic criterion that measures the deviation of the class distribution of the local pattern from the overall class distribution of the training examples. We will discuss such measures later on in this paper (Section 4). In the simplest case, local patterns are added until each example is covered by at least one local pattern.[1] Figure 1 shows the basic covering algorithm.

Rules are an obvious choice for local classifiers because a rule will typically only cover a subset of the entire example space. Consequently, rules are frequently used as a representation for local pattern discovery tasks such as association rule mining (Agrawal et al., 1995; Hipp et al., 2000) and subgroup discovery (Klösgen, 1996; Wrobel, 1997; Scheffer and Wrobel, 2002; Lavrač et al., 2004).

However, the covering algorithm does not depend on rule-based local patterns. Ferri et al. (2004) elegantly generalized the covering framework to arbitrary classifiers by defining the locality via a confidence threshold: The classifier is trained on all training examples, but it will not issue a prediction unless it has a certain (user-specified) minimum confidence in its prediction. All training examples that are classified with this

[1] In practice, this constraint is often relaxed to avoid overfitting.

```
function WeightedCovering(Examples)

# initialize classifier and example weights
GlobalClassifier $\leftarrow \emptyset$
foreach Example $\in$ Examples
    Weight(Example) $= 1.0$

# loop for a fixed number of iterations
for $i = 1 \ldots n$

    # find the best local pattern
    LocalPattern $\leftarrow$ FindBestLocalPattern(Examples)

    # add the local pattern to the classifier
    GlobalClassifier $\leftarrow$ GlobalClassifier $\cup$ LocalPattern

    # reduce the weight of covered examples
    ReduceWeights(Covered(LocalPattern,Examples))

return GlobalClassifier
```

Fig. 2. The Weighted Covering Algorithm

minimum confidence are then removed, and a new (possibly different type of) classifier is learned from the remaining examples.

A key problem for constructing a global theory out of local patterns is that the local patterns are discovered in isolation, whereas they will be used in the context of other patterns. The covering strategy partially addresses this problem by learning rules in order: all examples that are covered by previous patterns are removed from the training set before a new pattern is learned. This guarantees that the new local pattern will focus on new, unexplored territory. However, it also ignores the evidence contained in the removed examples, and the successive removal of training examples eventually leads to training sets with a very skewed class distribution, and possibly isolated, scattered examples.

As a remedy for this problem, several authors (Cohen and Singer, 1999; Weiss and Indurkhya, 2000; Gamberger and Lavrač, 2000) have independently proposed the use of *weighted covering* (Figure 2). The basic idea is to generalize the covering algorithm by introducing example weights. Initially, all examples have a weight of 1.0. However, the weights of examples that are covered by a rule will not be set to 0.0 (which is the equivalent to removing them from the training set), but instead their weight will only be reduced. This ensures that their influence on the evaluation of subsequent local patterns is reduced, but not entirely eliminated. Different algorithms use different weight adaptation formulas, ranging from error-based procedures motivated by boosting (Cohen and Singer, 1999) to simple techniques such as using the inverse of (one plus) the number of previous rules that cover the example (Gamberger and Lavrač, 2000).

Most weighted covering algorithms also adopt a very simple stopping criterion, namely they simply learn a fixed number of rules. Diversity of the rules is encouraged by the re-weighting of the examples, but it is no longer enforced that each example is covered by a rule. Also, the number of learned rules is typically higher, which has the effect that most examples will be covered by more than one rule. Thus, weighted covering algorithms have two complementary advantages: on the one hand they may learn better local pattern because the influence of previously covered patterns is reduced but they are not entirely ignored, on the other hand they will produce a better classifier by combining the evidence of more rules, thus exploiting the redundancy contained in an ensemble of diverse local patterns (Dietterich, 2000).

While covering and weighted covering try to take into account the context of previous patterns before generating a new local pattern, an alternative strategy is to try to "guess" what subsequent patterns may look like. One attempt into that direction is the PART algorithm (Frank and Witten, 1998), which does not learn the next local pattern in isolation but (conceptually) learns a global model in the form of a decision tree. From this tree, a single path is selected as the next local pattern that can be added to the theory.[2] In essence, this idea is a special case of the delegating classifiers framework discussed above (Ferri et al., 2004).

The best local patterns are typically found via a heuristic search, using some heuristic evaluation metric as a guide. We will discuss a few such measures further below, but first we we have to explain coverage spaces.

3 Coverage Spaces

In recent work, Fürnkranz and Flach (2005) introduced the framework of *coverage spaces* for analyzing rule evaluation metrics. Coverage spaces are a quite similar to ROC-spaces, the main differences being that coverage spaces work with absolute numbers of true positives and false positives (covered positive and negative examples), whereas ROC-spaces work with true positive and false positive rates. A rule (or a rule set) that covers p out of a total of P positive examples and n out of N negative examples is represented as a point in coverage space with the co-ordinates (n, p).

Adding a rule to a rule set increases the coverage of the rule set because an additional rule can only add new examples to the set of examples that are covered by 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 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. Figure 3 shows the coverage path for a theory with three rules. Each point R_i represents the rule set consisting of the first i rules. Adding

[2] The implementation of this phase can be optimized so that the selected branch can be grown directly, without the need of growing an entire tree first.

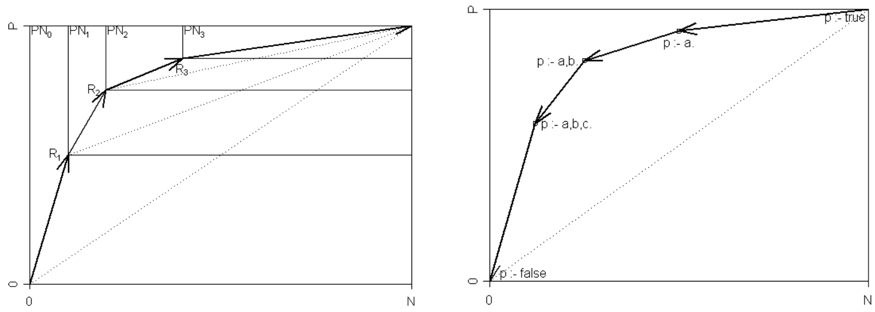


Fig. 3. Schematic depiction of the paths in coverage space for (left) the covering strategy of learning a rule set adding one rule at a time and (right) greedy specialization of a single rule.

a rule moves to a new point in coverage space, corresponding to a theory consisting of all rules that have been learned so far. Removing the covered examples has the effect of moving to a subspace of the original coverage space, using the last rule as the new origin. Thus the path may also be viewed as a sequence of nested coverage spaces PN_i. 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 $\tilde{R} = (N, P)$. Even this theory might be optimal under some cost assumptions.

For finding individual rules, the vast majority of algorithms use a heuristic top-down hill-climbing[3] or beam search strategy, i.e., they search the space of possible rules by successively specializing the current best rule (Fürnkranz, 1999). Rules are specialized by greedily adding the condition which promises the highest gain according to some *evaluation metric*. Just as with adding rules to a rule set, successive refinements of a rule describe a path trough coverage space (Figure 3, 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).

As we will see in the following, coverage spaces are well-suited for visualizing the behavior of evaluation metrics by looking at their *isometrics*, i.e., the lines that connect the rules that are evaluated equally by the used heuristic (Fürnkranz and Flach, 2005).

[3] If the term “top-down hill-climbing” sounds self-contradictory: hill-climbing refers to the process of greedily moving towards a (local) optimum of the evaluation function, whereas top-down refers to the fact that the search space is searched by successively specializing the candidate rules, thereby moving downwards in the generalization hierarchy induced by the rules.

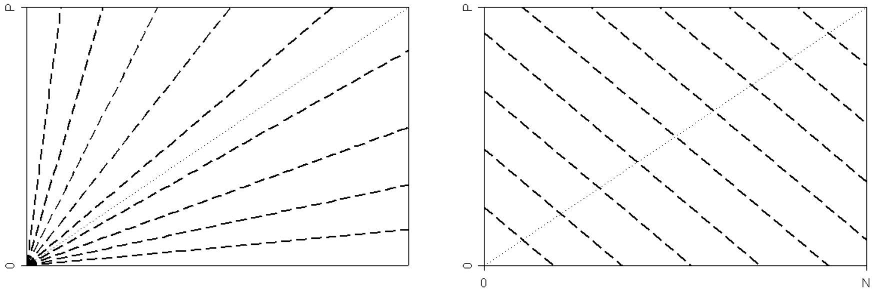


Fig. 4. Isometrics for precision gain (left) and coverage (right).

4 Rule Evaluation Measures

In each iteration, the covering algorithm needs to select the “best” local pattern that can be added. Informally, a good local pattern is a pattern for which the class distribution of the instances that it covers differs considerably from the overall class distribution. In a concept learning scenario (where we have only two classes, positive and negative examples for the target concept), we will try to identify regions of the instance space in which instances of the concept are denser than in the overall example distribution, i.e., in regions where there is a higher proportion of positive examples.

4.1 Trading off precision and coverage

Numerous rule evaluation measures have been proposed in various contexts (Fürnkranz, 1999; Lavrač et al., 1999; Tan et al., 2002). In the following, we will concentrate on a family of well-known evaluation metrics for subgroup discovery (Klösgen, 1992). They have in common that they trade off two basic components:

Precision Gain $g = \frac{p}{p+n} - \frac{P}{P+N}$ is the difference between the proportion of positive examples in the examples covered by the local pattern and the overall proportion of positive examples.

Coverage $c = \frac{p+n}{P+N}$ is the proportion of all examples that are covered by the local pattern.

Figure 4 shows the isometrics in coverage space for these two basic heuristics. Note that the second term of precision gain is constant for all local patterns. Thus, maximizing precision gain is the same as maximizing precision, and the isometric structure of precision gain is the same as the one for precision itself: The rules with the lowest evaluation are those on the N-axis because they only cover negative examples. Here, precision has its minimum value of 0 and precision gain the minimum value of $-P/(P+N)$. The examples with the highest evaluation can be found on the P-axis because those are the ones that cover only positives examples. There, precision has its maximal value of 1, and precision gain the maximum value of $1 - P/(P+N)$. In between, the isometrics rotate around the point $(0,0)$, the empty rule. For example, all rules on the diagonal (those for which the covered positives and negative examples are distributed in the same way as the examples in the overall distribution) are evaluated in the same way with this heuristic (with the value 0 in the case of precision gain and with $P/(P+N)$ for precision). The isometrics of coverage move in parallel lines from the empty rule (no coverage) to the universal rule (covering all examples). The lines have an angle of 45 degrees with the N- and P-axes because for coverage there is no difference in importance for covering a positive or covering a negative example.

Klösgen (1996) identified three different variations for combining these two measures, which satisfy a set of four basic axioms proposed by Piatetsky-Shapiro (1991) and Major and Mangano (1995). He further showed that several other measures are equivalent to these. Wrobel (1997) added a fourth version. All four measures only differ in the way in which they trade off coverage c versus precision gain g. These measures are:

(a) $\sqrt{c}g$ (b) cg (c) c^2g (d) $\frac{c}{1-c}g$

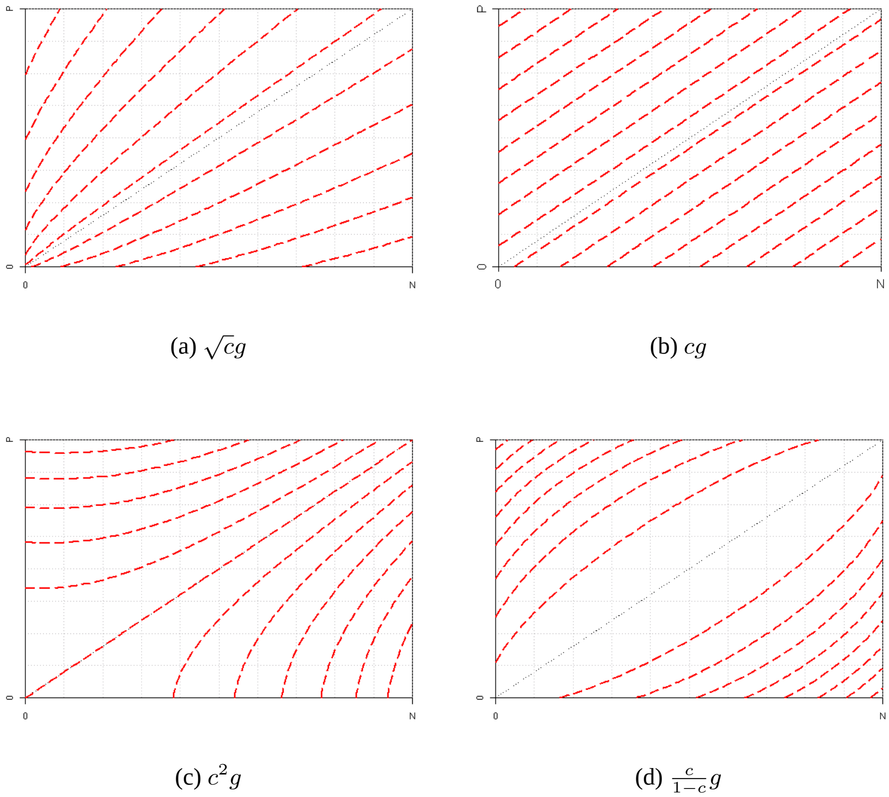
The isometrics of these measures are shown in Figure 5. Measure (a) was proposed by Klösgen (1992). Its idea is to perform a statistical test on the distribution of precision gain, under the assumption that, if the true precision of the rule were the same as the overall precision in the example set, the observed value for precision gain should follow a binomial distribution around 0. The variance of this distribution brings in the factor $\sqrt{c}$. The isometrics show that the measure has a slight tendency to prefer rules that are near the origin. In that region, the isometrics start to bend towards the origin, which means that rules with low coverage need smaller deviations from the diagonal than larger rules with the same evaluation.

Measure (b) is weighted relative accuracy, as proposed independently by Piatetsky-Shapiro (1991) and Lavrač et al. (1999). It has linear isometrics, parallel to the diagonal. Thus, all rules that have the same normal distance from the diagonal are evaluated in the same way, independent of their location in coverage space. In comparison to (a), this has increased the influence of coverage, with the result that smaller rules are no longer preferred.

Wrobel (1997) proposed to further strengthen the influence of coverage by squaring it, resulting in measure (c). This results in an isometric landscape that has a clear tendency to avoid the region with low coverage near the lower left corner (see Figure 5, lower left). Obviously, the rules found with this measure will have a stronger bias towards generality.

Klösgen (1992) has shown that measure (d) is equivalent to several other measures that can be found in the literature, including a χ^2-test. It is quite similar to the first measure, but its edges are bent symmetrically, so that rules with high coverage are penalized in the same way as rules with a comparably low coverage.

It is quite interesting to see that in regions with higher coverage, the isometrics of all measures except (d) approach parallel lines, i.e., with increasing rule coverage, they converge towards some measure that is equivalent to weighted relative accuracy. However, measures (a), (b), and (c) differ in their behavior near the low coverage region



(a) $\sqrt{c}g$

(b) cg

(c) c^2g

(d) $\frac{c}{1-c}g$

Fig. 5. Different ways of trading off coverage c and relative confidence g.

of the rule space. Measure (a) makes it easier for rules in the low-coverage region near the origin, (b) is neutral, whereas (c) penalizes this region.

It seems to be the case that two controversial forces are at work here: On the one hand, locality and coverage of a pattern are inversely correlated: the higher the coverage of a pattern, the more global the pattern. Thus, it seems reasonable to encourage the discovery of patterns with low coverage, as measure (a) does. On the other hand, low coverage patterns tend to be less reliable because their estimated performance parameters (such as their precision) are associated with a larger variance and a larger uncertainty. A simple, solution for this problem might be to try to avoid these regions if possible, as measure (c) does.[4] Weighted relative accuracy (b) tries to compromise between these two approaches. Note that precision may also be viewed in this framework, as giving no weight to the coverage of the rule (i.e., it is equivalent to c^0g).

[4] This is related to the *small disjuncts problem*: rules with high coverage are responsible for a large part of the overall error of a rule set. Nevertheless, the experiments in (Holte et al., 1989) suggest that avoiding them entirely is not a good strategy.

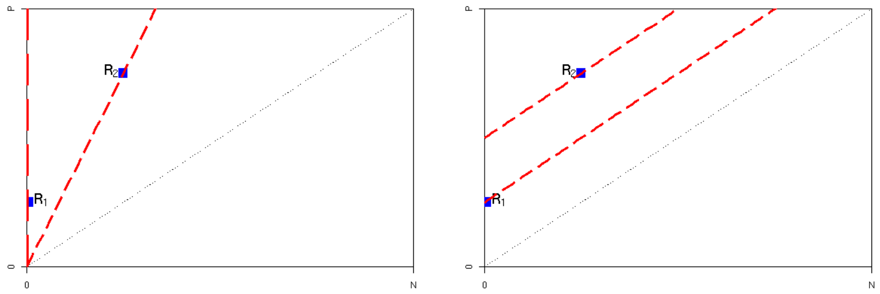


Fig. 6. Precision prefers the smaller, pure rule, whereas weighted relative accuracy prefers the larger rule with several exceptions

4.2 What is the Optimal Trade-off?

We have seen that all measures of the general form $c^w g$, for some $w \in \mathbb{R}, w \geq 0$, implement the basic idea of measuring the quality of a local pattern with the distance from the diagonal of the coverage space. The motivation for this approach is evident: the diagonal represents all rules that have the same overall distribution as can be found in the entire rule set, and the goal of local pattern discovery is to find a pattern that covers a set of instances that deviate significantly from this default distribution. The individual measures differ only in the way they measure this distance in different regions of coverage, i.e., in their different choices of w.

Consider the example shown in Figure 6. It shows two rules: R_1 is a pure rule, covering one fourth of all positive examples and no negative examples. R_2, on the other hand, covers $3/4$ of all positive examples but also a fourth of all negative examples, in a distribution where the prior probability of a positive example is $4/9$. Thus, the precision gain of R_1 is $1 - 4/9 = 5/9 = 0.444$, whereas the precision gain of R_2 is $\frac{3/4 \times 4/9}{3/4 \times 4/9 + 1/4 \times 5/9} - 4/9 = 12/17 - 4/9 = 0.261$. Clearly, rule R_1 is better according to this criterion, as can also be seen from the left graph in Figure 6.

On the other hand, if we evaluate with weighted relative accuracy, the picture changes: rule R_1 covers one fourth of all positive examples, i.e. $1/4 \times 4/9 = 1/9$ of all examples. Rule R_2, on the other hand, covers a $3/4$ of the positives, and $1/4$ of the negatives, in total $3/4 \times 4/9 + 1/4 \times 5/9 = 17/36$ of the total number of examples. Thus, weighted relative accuracy, which multiplies coverage with precision gain, yields $1/9 \times 1 = 1/9 = 9/81$ for rule R_1, and $17/36 \times (12/17 - 4/9) = 10/81$ for R_2. Note that these values are independent of the absolute number of examples that are covered by the rule, they only depend on the proportion of examples covered.[5]

[5] This will not change if absolute coverage instead of relative coverage is used in the formula because a multiplication with a constant $(P + N)$ will not change the isometric structure for any given coverage space.

However, intuitively, the validity and interestingness of the found patterns is not entirely clear. If rule R_1 covers only one or two positive examples, rule R_2 seems to be preferable because it is backed up with a larger amount of evidence and is therefore presumably more reliable. In our case, rule R_2 would cover about 3.5 examples, but it easy to construct examples where R_2 covers an arbitrary number of examples (increase the total number of examples and/or move the point upwards on its WRA isometric). On the other hand, if R_1 covers thousands of examples, a pure group of that size seems to be interesting irrespective of the total training set size.

Thus, we would propose that with growing coverage, coverage becomes less and less important for the evaluation of the quality of a found local pattern. In a crude form, this assumption can also be found in the support/confidence pruning framework that is paramount in association rule discovery: Rules below a given support threshold are not considered at all, rules above the given threshold are evaluated with their precision. Note, however, that support and coverage are not exactly the same: support is the proportion of *positive* examples that are covered.

This assumption may, of course, not hold for all applications. The choice we are making here is related to recall and precision trade-offs and misclassification costs, and clearly depends on the way the discovered patterns are put to use in a real-world application. However, for the case of discovering global patterns from local patterns, we believe that many pure rules are preferable to a few large, but impure rules, provided it is established that the precision estimate of the rule is valid. The latter, of course, is not true in typical rule learning applications, where the majority of the rules that are found with precision are rules covering only a few examples.

In any case, we believe that one of the main problems with the use of precision as a rule learning heuristic is that it is very susceptible to overfitting. Rules that cover only one or a few examples on the training set are evaluated with 100% precision, but their true precision in the entire domain will typically be much worse. Thus, many techniques have been proposed to make precision estimates more conservative, most prominently the Laplace- and m-estimates (Cestnik, 1990; Clark and Boswell, 1991). In the next section, we propose an alternative route that aims at the use of meta-learning for predicting the “true” precision of a rule.

5 Meta-learning Rule Precision

In this section, we discuss some experiments that aim at predicting the “true” precision of a rule that has been learned with a covering algorithm. Based on the rule’s coverage on the training set, we want to learn a function that predicts the rule’s precision on an independent test set. The basic idea is to generate a large number of rules, observe their precision on the training set and and independent test set, and learn a function that predicts the test set precision from the (absolute) number of covered examples on the training set. More details on this work can be found in (Fürnkranz, 2004a;b).

5.1 Meta Data Generation

We used a simple covering algorithm for learning a set of rules. For each learned rule, we recorded the numbers of covered positive and negative examples on both the training

```
procedure GenerateRules(TrainSet,TestSet)

# loop until all positive examples are covered
while Positive(TrainSet) $\neq \emptyset$

  # find the best rule
  Rule $\leftarrow$ GreedyTopDown(TrainSet)

  # stop if it doesn't cover more pos than negs
  if |Covered(Rule, Positive(Examples))|
    $\leq$ |Covered(Rule, Negative(Examples))|
    break

  # loop through all predecessors
  Pred $\leftarrow$ Rule
  repeat

    # record the training and test coverage
    TrainP $\leftarrow$ |Covered(Rule,Positive(TrainSet))|
    TrainN $\leftarrow$ |Covered(Rule,Negative(TrainSet))|
    TestP $\leftarrow$ |Covered(Rule,Positive(TestSet))|
    TestN $\leftarrow$ |Covered(Rule,Negative(TestSet))|
    print Pred,Rule,TrainP,TrainN,TestP,TestN

    Pred $\leftarrow$ RemoveLastCondition(Pred)
  until Pred = null

  # remove covered training and test examples
  TrainSet $\leftarrow$ TrainSet \ Covered(Rule,TrainSet)
  TestSet $\leftarrow$ TestSet \ Covered(Rule,TestSet)
```

Fig. 7. Covering algorithm for generating and evaluating rules

and an independent test set. We recorded these statistics not only for *final rules*—those rules that would be used in the final theory—but also for all their ancestors, i.e., for all *incomplete rules* that were eventually refined into a final rule. These can be simply obtained by deleting the final conditions of each rule. The main motivation for this step is that we want to have complete information on each path in the refinement graph that yields a final rule. Figure 7 shows the meta data generation algorithm in pseudo-code.

The algorithm for generating the individual rules is a straight-forward greedy top-down algorithm: rules are refined until no further refinement is possible. At each refinement step, all possible immediate refinements (adding one condition) are evaluated and the best one is selected. Among all rules encountered during this search, the best rule is eventually returned. Note, however, that the best rule need not be the last one searched.

We did not implement any method for pruning the obtained rules. Our main goal is to study the test set performance of individual rules, and not so much to learn a good theory. Therefore, the evaluation of possibly overfitting rules is very important to us. As a consequence, we chose not to implement any filtering heuristics which would prune those rules away. To ensure some variety in the size of the learned rules by using

Table 1. Search heuristics used in this study. p and n are the number of covered among a total of P and N positive and negative examples.

heuristic	formula
precision	$\frac{p}{p+n} \sim \frac{p-n}{p+n}$
Laplace	$\frac{p+1}{p+n+2}$
accuracy	$\frac{p+(N-n)}{P+N} \sim p - n$
weighted rel. acc.	$\frac{p+n}{P+N}(\frac{p}{p+n} - \frac{P}{P+N}) \sim \frac{p}{P} - \frac{n}{N}$
correlation	$\frac{p(N-n)-(P-p)n}{\sqrt{PN(p+n)(P-p+N-n)}}$

evaluation heuristics with very different biases (as will be explained below), some of which have a tendency to learn very general rules, while others are clearly prone to overfitting.

In order to collect statistics under a fairly broad set of conditions, we varied the following dimensions:

Datasets: We used 27 datasets with varying characteristics from the UCI repository. These datasets were selected because of their availability and moderate size. We did not include larger datasets (such as *shuttle*) because the region of interest (as we will see) is the region of rules with low coverage.

5x2 Cross-validation: For each dataset, we performed 5 iterations of a 2-fold cross-validation. 2-fold cross-validation was chosen because in this case the training and test sets have equal size, which makes a comparison of the obtained estimates easier. We collected statistics for all rules of all five iterations of two folds, i.e., a total of 10 per run.

Classes: For each dataset and each fold, we generated one dataset for each class, treating this class as the positive class and the union of all other classes as the negative class. Rules were learned for each of the resulting two-class datasets.

Heuristics: Finally, we ran the rule learner five times on each binary dataset, each time using a different search heuristic. We used the five heuristics shown in Table 1. The first four form a representative selection of search heuristics with linear isometrics (Fürnkranz and Flach, 2003), while the correlation heuristic (Fürnkranz, 1994) has non-linear isometrics. These heuristics represent a large variety of learning biases. For example, it is known that *WRA* and *Accuracy* tend to prefer simpler rules with high coverage, whereas *Precision* and *Laplace* tend to prefer possibly complex rules with high precision on the training set. Note that the correlation heuristic is equivalent to a χ^2-statistic (Fürnkranz and Flach, 2005), which in turn is equivalent to heuristic (d) of the previous section (Klösgen, 1992). We have not yet run experiments with heuristics (a) and (c).

In total, 5409 theories with 48,603 rules were learned. For all rules and ancestors we recorded their precision on the test set, resulting in statistics for a total of 114,375 rules. 13,399 rules did not cover any examples on the test set and were ignored. Our reasons for this were that on the one hand we did not have any training information for this rule (the test precision that we try to model is undefined for these rules), and that on the other hand such rules do not do any harm (they won't have an impact on test set accuracy as they do not classify any test example). Ignoring them seemed to be the most reasonable option for our purposes. The large majority of these ignored rules (9,806 rules) covered only a single positive and no negative examples on the training set. A total of 100,976 rules remained for the analysis.

Each rule is evaluated in the context of all previously learned rules, i.e., all examples covered by previous rules are removed from the dataset. Thus, later rules in a theory are learned from a smaller dataset than the first rules in the theory. This procedure was also mirrored in the test set. In other words, we assumed a decision list learning scenario, where an example is classified with the prediction of the first rule that fires on the example. Thus, rule n only receives examples (from both training or test sets) that are not covered by rules $1 \ldots n-1$.

5.2 Fitting Search Heuristics

We fitted several 2-dimensional functions to these meta data, with the goal of using them as a search heuristic inside the rule learner. We fitted the parameters of the following three types of heuristics:

- a neural network (fully connected with a five-node hidden layer, fitted using R's `nnet` procedure (Venables and Ripley, 2002))
- the m-estimate (Cestnik, 1990; Clark and Boswell, 1991), resulting in the function $\frac{p+1.6065*P/(P+N)}{p+n+1.6065}$ (fitted using R's `nls` procedure (Venables and Ripley, 2002))
- the generalized m-heuristic, which re-interprets the prior probability in the m-heuristic as a cost parameter (Fürnkranz and Flach, 2003), resulting in $\frac{p+0.785}{p+n+2.7153}$

The residual sum-of-squares showed the best fit for the m-heuristic ($rss = 7842.37$), followed by the neural network ($rss = 7897.1$) and the generalized m-heuristic ($rss = 8029.34$).

Table 2 shows the accuracy results (estimated by a 10-fold stratified cross-validation) for all eight heuristics[6] on the 27 data sets that were used for generating the meta data, as well as on 10 datasets that were not used in the training phase. At the bottom of Table 2, we also show the average rule sizes for each heuristic. As an independent benchmark, we also added the results of JRip, Weka's re-implementation of Ripper (Cohen, 1995), in two versions, without and with pruning. The results exhibit a fairly large variance. There are cases where the learned heuristics clearly outperform the five original heuristics (e.g., *labor*), but there are also cases where they are outperformed by at least one of them.

Table 3 shows the p-value of a paired t-test, and the number of wins and losses for each combination of a base heuristic with a meta-learned heuristic. It can be seen that the meta-learned heuristics outperform *Precision*, *Laplace*, and *Accuracy*. The differences for the neural network are not significant, but the trained m-estimates outperform

[6] The neural network was implemented via a look-up table of the average prediction values of 10 different networks for all combinations of values $n \leq 50$ and $p \leq 50$. Precision was used for all larger values.

Table 2. Accuracy and number of learned rules for the five basic and three learned heuristics on 10 new datasets. For comparison, we also show the result of JRip without Pruning (-P) and JRip, and the average results of the algorithms on the 27 datasets that were used for training.

	Prec	Lap	Acc	WRA	Corr	NNet	MEst	GenM	JRip -P	JRip
anneal	99.00	99.00	98.75	98.62	98.75	99.37	99.37	99.25	98.50	97.62
audiology	76.55	74.78	80.97	85.84	80.53	77.43	77.43	76.11	73.89	72.12
breast-cancer	68.88	67.48	72.73	69.93	66.78	68.88	65.03	70.63	73.43	72.38
cleveland-heart	72.61	70.96	73.60	72.28	76.90	72.28	72.61	74.92	77.56	79.54
contact-lenses	66.67	62.50	66.67	83.33	66.67	70.83	70.83	70.83	70.83	75.00
credit	84.69	84.90	84.90	86.73	83.88	85.51	83.27	83.67	85.31	85.71
glass	59.81	57.94	62.62	59.35	65.89	58.88	61.22	59.35	66.36	69.16
glass2	74.23	73.62	78.53	76.07	81.60	73.62	77.91	76.69	81.60	79.14
hepatitis	79.35	81.29	78.71	78.71	78.06	76.13	77.42	80.00	81.29	79.35
horse-colic	74.18	71.20	79.35	84.24	82.07	75.82	75.82	75.00	78.80	84.78
hypothyroid	97.91	98.17	98.13	98.32	98.77	98.58	98.42	98.70	98.70	99.11
iris	95.33	96.00	92.00	92.00	92.67	92.00	96.00	95.33	90.67	95.33
krkp	99.06	99.28	97.25	94.34	98.22	99.09	99.25	99.28	99.44	99.09
labor	87.72	87.72	85.96	82.46	87.72	89.47	91.23	89.47	84.21	77.19
lymphography	82.43	82.43	77.70	79.73	79.05	81.08	81.08	81.76	74.32	81.08
monk1	78.23	81.45	79.84	81.45	81.45	73.39	81.45	81.45	84.68	89.52
monk2	47.93	49.11	47.93	54.44	55.03	53.25	48.52	51.48	49.11	52.07
monk3	82.79	86.07	78.69	77.05	74.59	87.70	83.61	85.25	81.97	86.07
mushroom	100.00	100.00	98.23	96.45	98.23	100.00	100.00	100.00	100.00	100.00
sick-euthyroid	95.86	96.17	96.74	96.30	96.68	95.98	96.27	96.24	96.68	97.72
soybean	88.73	89.17	91.07	87.26	90.19	90.78	91.22	90.63	91.07	90.19
tic-tac-toe	97.29	97.29	88.41	71.40	83.09	97.29	97.08	97.29	97.18	97.18
titanic	78.33	78.33	78.33	77.60	77.78	78.33	78.33	78.33	78.33	78.24
vote	94.48	94.94	94.48	94.94	94.02	95.86	95.17	94.71	95.40	96.32
vote-1	88.97	87.13	88.51	89.66	90.11	89.20	87.36	89.20	88.28	88.97
vowel	50.10	50.71	47.37	63.03	70.81	54.14	55.35	52.22	74.75	72.63
wine	92.13	92.13	93.82	95.51	93.82	93.82	92.70	93.26	94.38	91.57
average (27 old)	81.97	81.84	81.90	82.48	83.09	82.55	82.74	83.00	83.95	84.71
balance-scale	73.44	73.12	68.80	66.88	77.76	71.84	72.32	72.80	80.32	81.28
breast-w	94.85	94.85	95.28	94.28	95.57	95.14	94.56	95.14	93.71	95.14
credit-g	69.10	70.00	67.00	72.50	69.60	67.70	68.10	69.20	73.30	70.80
diabetes	68.23	69.66	69.27	71.88	69.01	70.31	71.88	68.75	72.92	74.22
ionosphere	93.45	94.30	89.46	89.74	88.03	94.02	93.73	94.02	90.60	88.60
primary-tumor	33.04	32.74	29.50	35.40	35.40	33.63	34.81	33.33	39.23	38.94
segment	91.39	90.61	88.10	92.29	94.94	91.64	91.77	91.17	95.76	95.11
sonar	62.02	63.46	68.75	67.79	73.08	66.83	65.87	67.31	77.40	76.44
vehicle	69.39	67.14	62.65	60.52	68.44	65.48	71.63	67.49	67.02	68.68
zoo	84.16	85.15	90.10	92.08	90.10	89.11	90.10	90.10	87.13	86.14
average (10 new)	73.91	74.10	72.89	74.34	76.19	74.57	75.48	74.93	77.74	77.53
avg. # rules (27 old)	40.41	36.93	32.56	4.74	13.63	30.22	30.81	30.85	14.11	8.11
avg. # rules (10 new)	83.20	78.30	86.50	4.30	21.20	64.20	67.60	68.40	18.50	9.80

Table 3. Significance level of a paired t-test and number of wins and losses of pairwise comparisons between the base heuristics and the meta-learned heuristics.

	NNet	MEst	GenM
Precision	0.929 (10/23)	0.996 (9/25)	0.9996 (7/26)
Laplace	0.913 (10/23)	0.991 (12/21)	0.999 (10/22)
Accuracy	0.939 (13/21)	0.985 (13/22)	0.996 (12/23)
WRA	0.541 (20/15)	0.679 (16/19)	0.695 (16/19)
Correlation	0.178 (21/15)	0.291 (20/15)	0.321 (20/15)

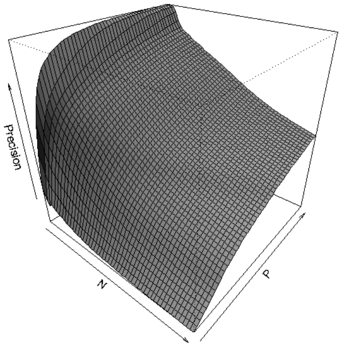


Fig. 8. Surface of a neural-net fit to the evaluation data

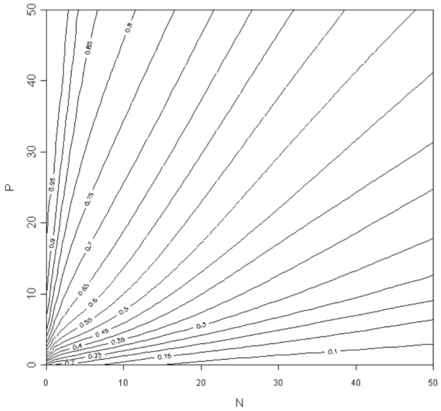


Fig. 9. Isometrics of a neural-net fit to the evaluation data

these heuristics in all but one case at the 1% significance level. On the other hand, weighted relative accuracy and correlation, which correspond to heuristics (b) and (d) of Figure 5, are *en par* with the meta-learned heuristics. These two differ from the others in that they are symmetrical around the diagonal, i.e., they incorporate information about the prior probability of the problem. Among the meta-learned heuristics, only the m-heuristic takes this information into account.

In particular, the performance of the neural network is somewhat disappointing. Although it is the most expressive model class (the only one that could be trained to fit non-linear isometrics), the net did not surpass the results of its linear competitors, not even at a significance level of 5%. Overfitting could be one cause, the above-mentioned absence of the prior probability as an additional input to the network another. Nevertheless, it is interesting to see how the network fitted the data. Figure 8 shows the surface of the learned evaluation function. Note that the steep non-linear shape for low levels of N and P gradually shifts towards an almost linear shape. This is not surprising, as the bias of the training set precision can be expected to be much lower for rules with high coverage than for rules with low coverage, because it is easier to fit a small sample by chance. Figure 9 shows the isometrics of the learned neural network. It is quite obvious that the shape is very similar to the shape of precision, which would be lines rotating around the angle $(0,0)$. However, while the lines become increasingly straight the farther they move away from the origin, they are quite non-linear near the origin. In these regions, it might make a difference whether a rule is evaluated with precision on the training set or with the predicted test set precision. Moreover, the isometrics do not meet in the origin, but seem to rotate around some point below it. This is characteristic of the m-estimate, and related heuristics. and may partly explain the good performance of such heuristics.

In general, our results are on average somewhat below those of JRip, although there are numerous exceptions. This difference could have several reasons, among them differences in implementation (all other algorithms differed only in the used heuristics, whereas JRip is a completely independent implementation) and the fact that our algorithms did not use any kind of noise handling. A somewhat unexpected side result of our experiments is that the no pruning version of JRip often outperforms its pruning counter-part (17 wins vs. 18 losses, with a p-value of 0.86). Thus, it can be assumed that the lack of a pruning option does not necessarily hamper the performance of our simple implementation of the separate-and-conquer algorithm on this selection of datasets.

6 Conclusions

In our view, it is still a largely open question what functions should be used to evaluate individual rules or candidate rules in a covering algorithm. The same holds for local patterns in general. In this paper, we have used the framework of coverage spaces to investigate a well-known family of evaluation metrics for subgroup discovery, which trades off coverage and precision of a rule. Our proposition is that for patterns with high coverage, coverage is of minor importance, and precision (or precision gain) should be used for evaluating the quality of the found pattern. The main motivation for including coverage seems to be to avoid overfitting, which made us investigate the possibility of meta-learning a function for predicting the "true" value of the precision of a rule.

Our empirical results show that meta-learning will yield improved results over several commonly used evaluation metrics. However, two of the measures that were originally proposed for subgroup discovery were *en par* with the meta-learned heuristics. We take this as evidence that in future work, we should (a) concentrate on investigating the quality of this family of subgroup discovery measures for inductive rule learning, and (b) repeat some of our meta-learning experiments with this function family. In particular, we plan to use a meta-learning approach like the one reported in this paper for fitting the parameter w of the function family $c^w g$. A key difference is that these heuristics (and the m-estimate, which also performed quite well) take the class distribution of the learning problem into account. We expect that including this as a third parameter would also improve the results of the fitted neural network. Note that such a procedure would effectively move operation into 3D-ROC space (Flach, 2003).

Acknowledgments

I would like to thank the participants of the Dagstuhl workshop on Local Patterns for enlightening discussions on the subjects of this paper. I would also like to thank the many people that make great software freely available. Weka, R, Perl and Cygwin were invaluable for this work.

References

R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. I. Verkamo. Fast discovery of association rules. In U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, pages 307–328. AAAI Press, 1995.

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.

P. Clark and R. Boswell. Rule induction with CN2: Some recent improvements. In *Proceedings of the 5th European Working Session on Learning (EWSL-91)*, pages 151–163, Porto, Portugal, 1991. Springer-Verlag.

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

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

T. G. Dietterich. Ensemble methods in machine learning. In J. Kittler and F. Roli, editors, *First International Workshop on Multiple Classifier Systems*, pages 1–15. Springer-Verlag, 2000.

C. Ferri, P. Flach, and J. Hernández. Delegating classifiers. In R. Greiner and D. Schuurmans, editors, *Proceedings of the 21st International Conference on Machine Learning (ICML-04)*, pages 289–296, Sydney, Australia, 2004. Omnipress.

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

E. Frank and I. H. Witten. Generating accurate rule sets without global optimization. In J. Shavlik, editor, *Proceedings of the 15th International Conference on Machine Learning (ICML-98)*, pages 144–151, Madison, Wisconsin, 1998. Morgan Kaufmann.

J. Fürnkranz. FOSSIL: A robust relational learner. In F. Bergadano and L. De Raedt, editors, *Proceedings of the 7th European Conference on Machine Learning (ECML-94)*, pages 122–137, Catania, Italy, 1994. Springer-Verlag.

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

J. Fürnkranz. Modeling rule precision. In J. Fürnkranz, editor, *Proceedings of the ECML/PKDD-04 Workshop on Advances in Inductive Rule Learning*, pages 30–45, Pisa, Italy, 2004a.

J. Fürnkranz. Modeling rule precision. In A. Abecker, S. Bickel, U. Brefeld, I. Drost, N. Henze, O. Herden, M. Minor, T. Scheffer, L. Stojanovic, and S. Weibelzahl, editors, *Lernen – Wissensentdeckung — Adaptivität. Proceedings of the LWA-04 Workshops*, pages 147–154, Humboldt-Universität zu Berlin, 2004b.

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

J. Fürnkranz and P. Flach. ROC 'n' rule learning – Towards a better understanding of covering algorithms. *Machine Learning* 58, 2005. In press.

D. Gamberger and N. Lavrač. Confirmation rule sets. In D. A. Zighed, J. Komorowski, and J. Żytkow, editors, *Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD-00)*, volume 1910 of *Lecture Notes in Artificial Intelligence*, pages 34–43, Lyon, France, September 2000. Springer, Berlin.

J. Hipp, U. Güntzer, and G. Nakhaeizadeh. Algorithms for association rule mining – a general survey and comparison. *SIGKDD explorations*, 2(1):58–64, June 2000.

R. Holte, L. Acker, and B. Porter. Concept learning and the problem of small disjuncts. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence (IJCAI-89)*, pages 813–818, Detroit, MI, 1989. Morgan Kaufmann.

W. Klösgen. Problems for knowledge discovery in databases and their treatment in the statistics interpreter EXPLORA. *International Journal of Intelligent Systems*, 7(7): 649–673, 1992.

W. Klösgen. Explora: A multipattern and multistrategy discovery assistant. In U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, chapter 10, pages 249–271. AAAI Press, 1996.

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

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

J. A. Major and J. J. Mangano. Selecting among rules induced from a hurricane database. *Journal of Intelligent Information Systems*, 4(1):39–52, 1995.

G. Piatetsky-Shapiro. Discovery, analysis, and presentation of strong rules. In G. Piatetsky-Shapiro and W. J. Frawley, editors, *Knowledge Discovery in Databases*, pages 229–248. MIT Press, 1991.

T. Scheffer and S. Wrobel. Finding the most interesting patterns in a database quickly by using sequential sampling. *Journal of Machine Learning Research*, 3:833–862, 2002.

P.-N. Tan, V. Kumar, and J. Srivastava. Selecting the right interestingness measure for association patterns. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-02)*, pages 32–41, Edmonton, Alberta, 2002.

W. N. Venables and B. D. Ripley. *Modern Applied Statistics with S*. Springer, fourth edition, 2002.

S. M. Weiss and N. Indurkhya. Lightweight rule induction. In P. Langley, editor, *Proceedings of the 17th International Conference on Machine Learning (ICML-2000)*, pages 1135–1142, Stanford, CA, 2000.

S. Wrobel. An algorithm for multi-relational discovery of subgroups. In *Proceedings of the First European Symposion on Principles of Data Mining and Knowledge Discovery (PKDD-97)*, pages 78–87, Berlin, 1997. Springer-Verlag.