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 parameterized 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.

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, including RIPPER, 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, 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) \( h_{\text{tpr}} = h_{\text{Recall}} = \frac{p}{P} \)

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.

\(^1\) As longer rules typically cover fewer examples, we would argue that this is just another way of measuring coverage. Also, in [8] it was recently found that including rule length does not improve the performance on heuristics that have been derived by meta-learning.