Resolving Rule Conflicts with Double Induction

Tony Lindgren and Henrik Boström

Department of Computer and Systems Sciences,
Stockholm University and Royal Institute of Technology,
Forum 100, 164 40 Kista, Sweden
{tony, henke}@dsv.su.se

Abstract. When applying an unordered set of classification rules, the rules may assign more than one class to a particular example. Previous methods of resolving such conflicts between rules include using the most frequent class in the conflicting rules (as done in CN2) and using naïve Bayes to calculate the most probable class. An alternative way of solving this problem is presented in this paper: by generating new rules from the examples covered by the conflicting rules. These newly induced rules are then used for classification. Experiments on a number of domains show that this method significantly outperforms both the CN2 approach and naïve Bayes.

1 Introduction

Two major induction strategies are used in top-down rule induction: Divide-and-Conquer [9] and Separate-and-Conquer (SAC) [5]. The former strategy does not generate overlapping rules and hence no conflict between rules can occur, while the latter strategy can give rise to overlapping rules. The rules can be ordered to avoid conflicts (which results in so called decision lists) [10] or they may be used without any ordering [3,2]. In the latter case one has to deal with conflicts that can occur when two or more rules cover the same example but assign different classes to it.

This work addresses the problem of handling conflicting rules and how to solve the conflicts in a more effective way than using existing methods. Previous methods that deals with conflicts among rules either returns the most frequent class covered by the conflicting rules (as done in CN2 [2]) here referred to as frequency-based classification, or uses naïve Bayes [8] to calculate the most probable class, as done in the rule induction system RDS [1].

We propose another way of resolving such conflicts: by applying rule induction on the examples covered by the conflicting rules, in order to generate a new set of rules that can be used instead of the conflicting rules. The motivation for this method is that by focusing on the examples within the region of interest (i.e. where the example to be classified resides), one is likely to obtain rules that better separate classes in this region compared to having to consider all examples, since examples outside the region may dominate the rule induction process, making the separation of classes within the region of marginal importance. This novel method is called Double Induction.
The paper is organized as follows. In section 2, we first review frequency-based classification and naïve Bayes to resolve rule conflicts, and then describe Double Induction. In section 3, the three different methods are compared empirically and the results are presented. Finally, in section 4, we discuss the results and give pointers to future work.

2 Ways of Resolving Classification Conflicts

In this section, we first recall two previous methods for resolving classification conflicts among overlapping rules and then introduce the novel method, Double Induction.

2.1 Frequency-Based Classification

The system CN2 [2] resolves classification conflicts between rules in the following way. Given the examples in Fig. 1, the class frequencies of the rules that covers the example to be classified (marked with ‘?’) are calculated:

\[
C(+) = \text{covers}(R_1, +) + \text{covers}(R_2, +) + \text{covers}(R_3, +) = 32
\]

and

\[
C(-) = \text{covers}(R_1, -) + \text{covers}(R_2, -) + \text{covers}(R_3, -) = 33
\]

where \(\text{covers}(R, C)\) gives the number of examples of class \(C\) that are covered by rule \(R\). This means that CN2 would classify the example as belonging to the negative class (−). More generally:

\[
\text{FreqBasedClassification} = \arg\max_{\text{Class}_i \in \text{Classes}} \sum_{j=1}^{\left|\text{CovRules}\right|} \text{covers}(R_j, C_i)
\]

where \(\text{CovRules}\) is the set of rules that cover the example to be classified.

2.2 Naïve Bayes Classification

Bayes theorem is as follows:

\[
P(C|R_1 \land \ldots \land R_n) = P(C) \frac{P(R_1 \land \ldots \land R_n|C)}{P(R_1 \land \ldots \land R_n)}
\]

where \(C\) is a class label for the example to be classified and \(R_1 \ldots R_n\) are the rules that cover the example. As usual, since \(P(R_1 \land \ldots \land R_n)\) does not affect the relative order of different hypotheses according to probability, it is ignored. Assuming (naively) that \(P(R_1 \land \ldots \land R_n|C) = P(R_1|C) \ldots P(R_n|C)\), the maximum a posteriori probable hypothesis (MAP) is:

\[
h_{\text{MAP}} = \arg\max_{\text{Class}_i \in \text{Classes}} P(\text{Class}_i) \prod_{R_j \in \text{Rules}} P(R_j|\text{Class}_i)
\]