Constructing Conjunctive Attributes Using Production Rules

Zijian Zheng
School of Computing and Mathematics
Deakin University, Geelong
Victoria 3217, Australia
Email: zijian@acm.org

Many existing constructive decision tree learning algorithms such as Fringe and Citre construct conjunctions or disjunctions directly from paths of decision trees. This paper investigates a novel attribute construction method for decision tree learning. It creates conjunctions from production rules that are transformed from decision trees. Irrelevant or unimportant conditions are eliminated when paths are transformed into production rules. Therefore, this new method is likely to construct new attributes with relevant conditions. Three constructive induction algorithms based on this basic idea are described and are empirically evaluated by comparing with C4.5 and a Fringe-like algorithm in a set of artificial and natural domains. The experimental results reveal that constructing conjunctions using production rules can significantly improve the performance of decision tree learning in the majority of the domains tested in terms of both higher prediction accuracy and lower theory complexity. These results suggest an advantage of the attribute construction method that uses production rules over the method of constructing new attributes directly from paths in noisy domains.

Keywords: Constructive Induction, Decision Tree Learning, Classification, Machine Learning.

CR Categories: Computing Methodologies: Artificial Intelligence (Subarea: Machine Learning).

1. INTRODUCTION

A well-known fundamental limitation of *selective induction* algorithms is that when task-supplied attributes are not adequate for describing theories to be learned, their performance in terms of prediction accuracy and theory complexity is usually poor. The *replication problem* (Pagallo and Haussler, 1990; Pagallo, 1990) of decision trees (Quinlan, 1993; Breiman, Friedman, Olshen and Stone, 1984) is a manifestation of this fundamental limitation of selective induction. Since a decision tree divides an instance space into mutually exclusive regions to represent a concept, a tree may contain duplication of a sequence of tests in different paths such as that shown (grey parts) in Figure 1. The replication leads to inconcise representations. If a subtree is duplicated several times in a tree, a certain number of training examples are needed to grow each of them. Otherwise the

Manuscript received: July 1997
Associate Editor: Maria Orlowska