ABSTRACT

Decision tree induction algorithms are highly used in a variety of domains for knowledge discovery and pattern recognition. They have the advantage of producing a comprehensible classification model and satisfactory accuracy levels in several application domains. Most well-known decision tree induction algorithms perform a greedy top-down strategy for node partitioning that may lead to sub-optimal solutions that overfit the training data. Some alternatives for the greedy strategy are the use of ensemble of classifiers or, more recently, the employment of the evolutionary algorithms (EA) paradigm to evolve decision trees by performing a global search in the space of candidate trees. Both strategies have their own disadvantages, like the lack of comprehensible solutions (in the case of ensembles) or the high computation cost of EAs. Hence, the authors of this chapter present a new algorithm that seeks to avoid being trapped in local-optima by doing a beam search during the decision tree growth. In addition, their strategy keeps the comprehensibility of the traditional methods and is much less time-consuming than evolutionary algorithms.
INTRODUCTION

Decision tree induction has been widely applied to a broad range of areas, such as medical diagnosis and assessment of credit risk. The induction of optimal decision trees, however, has been proven to be NP-Hard (Tan et al., 2005). Consequently, heuristics methods are required for solving the problem.

There is a clear preference in the literature for algorithms that rely on a greedy, top-down, recursive partitioning strategy for the growth of the tree. These algorithms use variants of impurity measures like information gain (Quinlan, 1986), gain ratio (Quinlan, 1993), gini-index (Breiman et al., 1984), distance-based measures (Mántaras, 1991), etc.

The greedy partitioning strategy has the advantage of having inexpensive cost, producing decision trees quite rapidly, even for large data sets. Notwithstanding, it presents two major drawbacks: (i) produces locally (rather than globally) optimal solutions, (ii) iteratively degrades the quality of the data set for the purpose of statistical inference, because the larger the number of times the data is partitioned, the smaller the data sample that fits the current split becomes, making such results statistically insignificant and thus contributing to a model that overfits the data.

Two main threads in the literature addressed the mentioned problems. The first one focused on building ensemble methods (see, for instance, Quinlan (1996)). These methods aim at building different decision trees by sampling the training data and using a majority voting scheme to decide the classification. A downside of this approach, from the user’s perspective, is that the simplicity of analyzing a single decision tree is lost. The second thread focused on using evolutionary algorithms to evolve decision trees in an attempt to find globally optimal solutions (see Barros et al. (2010) and Basgalupp et al. (2009) for successful examples). Nonetheless, evolutionary algorithms are time-consuming, mainly because the fitness component has to evaluate repeatedly the set of candidate solutions. Fitness evaluation is particularly costly when dealing with classifiers (more details in Espejo et al. (2010)).

In this work, we present a different approach for building decision trees. Instead of a greedy search, we use a beam search method in an attempt to avoid being trapped in local-optimas. Beam search is a more efficient version of the well-known best-first search that reduces its memory requirements. Instead of keeping track of all possible states, beam search stores a predetermined number of states, namely the beam width (w), according to their heuristic values. Actually, beam search with w = 1 is equivalent to the greedy search, and w = \infty is equivalent to breadth search.

Our new approach, namely Beam Classifier, uses a fixed beam width to reduce the chance of getting trapped into local optima. In addition, our strategy keeps the advantage of producing a single decision tree (easier to be interpreted than an ensemble of trees) and is much less time-consuming than evolutionary algorithms. Experimentation shows that our approach presents good results in several public data sets.

This work is organized as follows. We review classification and decision tree induction methods and then we detail our approach. Next, we present our experiments and we conclude this work with a summarized discussion of the results and future work.

CLASSIFICATION

Classification aims at building a concise class distribution model taking into account a set of predictive attributes. The outcome of such a model is used for assigning class labels to new examples whose only known information are the values of the predictive attributes (Ye, 2003).

The set of records whose class distribution is known is called the training set, and it can be described by a set of examples of the form (X, y),

358