A Decision-Tree Framework for Instance-space Decomposition

Shahar Cohen\textsuperscript{1}, Lior Rokach\textsuperscript{2}, and Oded Maimon\textsuperscript{1}

\textsuperscript{1} Department of Industrial Engineering, Tel-Aviv University, Israel
\texttt{shaharco@post.tau.ac.il, maimon@eng.tau.ac.il}
\textsuperscript{2} Department of Information Systems Engineering, Ben-Gurion University of the Negev, Israel \texttt{liorrk@bgu.ac.il}

\textbf{Summary.} This paper presents a novel instance-space decomposition framework for decision trees. According to this framework, the original instance-space is decomposed into several subspaces in a parallel-to-axis manner. A different classifier is assigned to each subspace. Subsequently, an unlabelled instance is classified by employing the appropriate classifier based on the subspace where the instance belongs. An experimental study which was conducted in order to compare various implementations of this framework indicates that previously presented implementations can be improved both in terms of accuracy and computation time.

\section{1 Introduction}

The multiple-classifier approach may improve the performance of a certain classification method. Both decomposition methodology as well as ensemble methodology applies a multiple-classifier approach for solving a classification task. Nevertheless the main idea of ensemble methodology is to combine a set of classifiers, each of which solves the same original task, in order to obtain a more accurate and reliable result than using a single classifier [4]. In a typical ensemble setting, each classifier is trained on data taken or re-sampled from a common dataset. On the other hand, the purpose of decomposition methodology is to break down a complex problem into several manageable problems, and to let each classifier solve a different task, i.e., individual classifiers cannot provide a solution to the original task.

Instance-space decomposition is a specific decomposition approach in which the original instance-space is decomposed into several subspaces. For each subspace, a different classifier is generated. Subsequently an unlabelled instance can be classified by selecting the appropriate classifier according to the subspace to which the instance belongs.

Several researchers proposed methods for combining, selecting and weighting the classification results of a set of given classifiers [5]. These methods are
based on evaluating the level to which a certain classifier is appropriate for a certain region in the instance-space.

NBTree is an instance space decomposition method that induces a decision tree and a naïve-Bayes hybrid classifier [3]. Naïve-Bayes, which is a classification algorithm based on Bayes’ theorem and a naïve independence assumption, is very efficient in terms of its processing time. To induce an NBTree, the instance space is recursively partitioned according to attributes values. The result of the recursive partitioning is a decision tree whose terminal nodes are naïve-Bayes classifiers. Since subjecting a terminal node to a naïve-Bayes classifier means that the hybrid classifier may classify two instances from a single hyper-rectangle region into distinct classes, the NBTree is more flexible than a pure decision tree. In order to decide when to stop the growth of the tree, NBTree compares two alternatives in terms of error estimation - partitioning into a hyper-rectangle regions and inducing a single naïve-Bayes classifier. The error estimation is calculated by cross-validation, which significantly increases the overall processing time. Although NBTree applies a naïve-Bayes classifier to decision tree terminal nodes, classification algorithms other than naïve-Bayes are also applicable. However, the cross-validation estimations make the NBTree hybrid computationally expensive for more time-consuming algorithms such as neural networks.

Although different researchers have addressed the issue of instance space decomposition, there is no research that suggests an automatic procedure for mutually exclusive instance space decompositions, which can be employed for any given classification algorithm and in a computationally efficient way. This paper introduces a decision-tree framework for instance-space decomposition (DFID). This framework grows a decision tree and applies a certain induction algorithm to its terminal nodes. While growing the tree, DFID can use several splitting criteria. Nevertheless a new splitting criterion that combines gain ratio measure with a subset grouping procedure is suggested, and its contribution to the problem is discussed. The DFID implementation with the new splitting criterion is called CPOM (Contrasted POpulations Miner).

The proposed algorithm can be used for developing lookahead based algorithms for induction of decision trees. Lookahead based algorithms attempts to predict the profitability of a split at a node by estimating its effect on deeper descendents of the node [6, 2]. For this purpose, one can use the DFID while employing a decision tree induction algorithm as the inner inducer.

2 Decision-tree Framework for Instance-space Decomposition (DFID)

Given a training set S drawn from a distribution D over the labelled instance space U, and an inducer I, the aim’ of the instance-space decomposition problem is to find an optimal decomposition W such that a unique classifier is derived for each of the regions and the sum of generalization errors over the