Extending the Learnability of Decision Trees

(extended abstract)

Tapio Elomaa

Department of Computer Science, University of Helsinki
Teollisuuskatu 23, SF-00510 Helsinki, Finland
elomaa@ca.helsinki.fi

1 Polynomial learnability of decision trees

Valiant's [9] pac learning framework and its subsequent elaborations deal with Boolean learning domains. Multivalued descriptor variables, or attributes, are easily dealt within the basic model [1]. However, learning functions that are not Boolean ones (i.e., that have a multivalued range) requires an extended model. Several such extensions have been put forward recently. We concentrate on Natarajan's [6] framework for learning classes of total functions on discrete domains.

In principle a multiclass problem can be handled by Boolean methods: Consider instances of one class (concept) as positive examples and the instances of all other classes as negative examples. Repeat the learning procedure for every class. The result is a separate hypothesis for every concept. In practice this approach has many difficulties. Most importantly, related concepts typically share many properties so that space and time can be saved by learning a common representation, a classifier, for all of them. Furthermore, categorizing instances using a classifier is less time-consuming than applying distinct hypotheses.

Decision tree learning is one of the most extensively studied problems of practical machine learning research. On the theoretical side, Ehrenfeucht and Haussler [3] have shown that a subclass of decision trees is learnable in the sense of Valiant. We generalize their definitions to m-ary domains and show that learnability of restricted decision tree classifiers carries over to the extended model.

A m-ary decision tree over n variables is a positional tree in which each internal node has exactly m sons and is labeled by one of the m variables. Each leaf is labeled by a value in \{1, 2, ..., m\}. A (m-ary) decision tree \( T \) defines a mapping \( f_T \) from \{1, 2, ..., m\} to \{1, 2, ..., m\} in a natural way. A tree is reduced if each variable appears at most once in a path from the root to a leaf.

The rank of a reduced decision tree \( T \), denoted \( r(T) \), is defined as follows:

1. If \( T \) consists of a single leaf node, then \( r(T) = 0 \).
2. Else if \( T_{\text{max}} \) is a subtree of \( T \) with the maximum rank \( r_{\text{max}} \), then
   \[
   r(T) = \begin{cases} 
   r_{\text{max}}, & \text{if } T_{\text{max}} \text{ is unique;} \\
   r_{\text{max}} + 1, & \text{otherwise.}
   \end{cases}
   \]

A combinatorial calculation shows that the number of decision trees of rank at most \( r \) over \( n \) m-ary variables, \( F(n, m, r) \), is as follows:

**Lemma 1** (i) Let \( k \) be the number of nodes in a reduced m-ary decision tree of rank \( r \) over \( n \) variables, where \( n \geq r \geq 1 \) and \( m \geq 2 \). Then \( 2^r m - m + 1 \leq k \), and

\[
k \leq (m - 1)^{-1} \left( m \sum_{i=0}^{r} \binom{n}{i} (m - 1)^i - 1 \right)
\]

\[
< 2e(mn/r)^r,
\]

where \( e \) is the base of the natural logarithm.

(ii) If \( r = 0 \) then \( F(n, m, r) = m \), else \( F(n, m, r) = m^{m^n} \) if \( n \leq r \), and \( F(n, m, r) \leq (4mn)^{e(mn/r)^r} \) if \( n > r \).

From this result it follows that in the interesting case, when \( n > r \), the class of m-ary decision trees of rank at most \( r \) over \( n \) variables is polynomial-sized for fixed \( m \) and \( r \); i.e., \( \ln(F(n, m, r)) \leq (e(mn/r)^r) \ln(4mn) \), which is polynomial in \( n \).

Ehrenfeucht and Haussler's [3] algorithm FINDMIN for identifying consistent decision trees of minimum rank is easily modified to identify m-ary decision trees of minimum rank. The running time remains polynomial in the sample size. This immediately gives us a way of constructing a probably approximately correct learning algorithm running in time polynomial in the number of attributes, \( n \), the inverse of the accuracy parameter, \( \epsilon^{-1} \), and the inverse of the confidence parameter, \( \delta^{-1} \), for fixed \( r \). Now, by Natarajan's [6] results, this together with Lemma 1 suffices to prove the learnability of m-ary decision trees of bounded rank in the extended model and we have the following result:

**Theorem 2** The class of functions represented by m-ary decision trees of fixed rank is polynomial-time learnable.
2 Adding heuristic techniques

The FINDMIN algorithm as such is not well-suited for practical purposes, despite its provably good behaviour. Its applicability can be increased by adding heuristic techniques, familiar from other decision tree algorithms, into it.

In empirical decision tree learning algorithms split attribute selection has proven to be of major importance. On the other hand, there are well-known difficulties associated with the information theoretic heuristics commonly used in empirical algorithms. In FINDMIN only one requirement is posed to an attribute to be placed at an internal node: It has to distinguish between at least two instances. It would seem beneficial to apply a heuristic technique here to improve FINDMIN's performance.

Mutikainen [5] has conducted a study on the effects of adding a heuristic attribute selection criterion to FIND, which is the only subprogram of FINDMIN. This and other minor modifications can be done without compromising the theoretical results. One such minor modification is handling attributes of varying arity. Then, in the theoretical results, \( m \) denotes the maximum arity of any attribute.

Mutikainen observed that employing an information theoretic attribute selection technique constantly decreased the size of the decision trees produced by FINDMIN. Slightly increased execution time, due to information theoretic merit calculations, follows. The asymptotic time requirement of FINDMIN does not, however, change.

We have contrasted FINDMIN enhanced with a heuristic attribute selection criterion against two practical induction algorithms in several real-life domains. The empirical algorithms are an implementation of ID3 decision tree learning algorithm [7] and the latest version of CN2 decision rule learning algorithm [2]. Our experience is that the enhanced FINDMIN proves to be competitive in performance with the practical methods, improves significantly on the performance of the basic FINDMIN, and is able to avoid some of the difficulties associated with information theoretic attribute selection functions, because FINDMIN uses the heuristic technique only to aid search primarily based on the minimum rank property.

With respect to ID3 and CN2's performance FINDMIN does not quite achieve the prediction accuracy of CN2 (80.6% vs. 82.7% average accuracy over eight domains), but clearly outperforms ID3 (76.8%) in this respect. FINDMIN also produces significantly smaller trees than ID3, and these trees have lower rank than those produced by ID3.

Tolerating noisy data, in addition to a successful attribute selection criterion, is an important aspect of empirical decision tree learning algorithms. The resulting tree is usually made robust against unsystematic errors by pruning. After growing a tree, it is pruned back to reduce its dependency on the specialities of the training data.

We have experimented with adding a heuristic pruning method into FINDMIN. This, unfortunately, does not preserve the formal learnability results. It is just recently that the learnability of decision trees of bounded rank in the presence of classification noise was independently proved by Elomaa and Kivinen [4] and Sakakibara [8]. This result is obtained by furnishing Ehrenfeucht and Haussler's FIND procedure with a technique that resembles pruning in practical decision tree algorithms. The same technique can be employed in the modified FIND procedure. Thus, the learnability of m-ary decision trees in the presence of classification noise can be proved.

Our experiences with adding a heuristic pruning method into FINDMIN are not as promising as our results with adding the attribute selection technique into FINDMIN. In artificially corrupted domains the overall performance of FINDMIN collapsed to the level of ID3's performance, and CN2 outperformed both these decision tree learning algorithms with a clear margin. The failure of FINDMIN is partly due to the insufficiency of the selected pruning technique, but probably also partly due to that decision trees of minimum rank do not tolerate any further processing.

With the formal learnability results for classification noise affected domains at hand the most interesting future research direction is to examine the applicability of the new modified algorithm to real-life problems.

Acknowledgments

This work was supported by the Academy of Finland.

References


