# Hybrid Classification: Using Axis-Parallel and Oblique Subdivisions of the Attribute Space (Extended Abstract)

Barbara Schulmeister and Mario Bleich

Fraunhofer Institute for Information and Data Processing
Branch Lab for Process Optimisation
Kurstraße 33, D-10117 Berlin, Germany
email: schulmei/bleich@epo.iitb.fhg.de

## 1 Introduction and Motivation

Each individual algorithm for supervised concept learning has advantages and disadvantages. This implies that no learning formalism can be the best for solving all classification tasks. The paper presents a hybrid algorithm which uses the strengths of standard decision tree algorithms and piecewise linear classifiers because at every level of learning it chooses the appropriate subdivision of the attribute space: a split with hyperplanes in general position or an axis-parallel split. Most of the decision tree algorithms can split the attribute space in axis-parallel hyperrectangles only, especially all the well-known, intensively studied and used algorithms (ID3, CART, IndCART, C4.5), initially introduced in domains with categorial attributes and later extended to numeric attributes. There are some often repeated important advantages of application of these decision tree algorithms.
But it is clear, that when the underlying concept is defined by hyperplanes in general position in the attribute space, axis-parallel splitting methods have to produce many decision nodes for the same attributes, the resulting trees are very large and generalize poorly for unobserved patterns.
Statistical methods and neural nets can produce good classifiers in such cases, but they are more concerned with performance as measured by error rate than with interpretability of the detected concept.
Another point, decision tree algorithms work well in comparison with classical statistical methods when the data are multimodal.
However, an important criterion for a classification method to qualify under the machine learning heading is that the derived rules should be meaningful to humans and evaluable in the head. On this basis, many statistical and neural net algorithms (and so the piecewise linear classifier DIPOL which is used in combination with a decision tree algorithm) would not qualify.
In this sense the introduced algorithm makes a good compromise between interpretability, compactness, and correctness of the learned concept.
There are some other developments in this direction. In the last years decision tree algorithms have been studied in which boolean combinations of attributes and more and more general combinations are applied for the split at a node.

A comprehensive review can be found in [4]. The so-called oblique decision tree algorithms use general linear combinations of the attributes and were suggested in the book of Breiman et al. [1] for the first time, but there has been only little further work on such trees until relatively recently. The successors are linear machine decision trees [6], [7]. Two other approaches use randomizing to find good oblique splits and to overcome the computational complexity of this problem: Simulated Annealing of Decision Trees (SADT, [2]) and Oblique Classifier 1 (OC1, [4]).

## 2 Description of the Hybrid Algorithm DIPOL-DT

The hybrid algorithm DIPOL-DT combines the piecewise linear classifier DIPOL and the decision tree algorithm CAL5. The two following subsections describe these algorithms developed in the authors department. Suppose that a finite set of classified examples with real-valued attributes is given for learning.

### 2.1 The Piecewise Linear Classifier DIPOL

DIPOL is a learning algorithm which constructs an optimized piecewise linear classifier for $n$-class problems [3].

- In the first step of the algorithm, initial positions of the discriminating hyperplanes are determined by linear regression for each pair of classes. It is well-known, that there is no guarantee to find a separating hyperplane with linear regression in the separable case. The reason of this can be found in the fact that this solution puts the emphasis on regions with high pattern density more than on the boundary region. Because of that
- the positions of the hyperplanes are optimized in the second step of the algorithm. An error criterion function is defined depending on the misclassified patterns. This function is minimized by a gradient descent procedure for each hyperplane separately. Each newly generated weight vector is compared to the existing, and only if the criterion function is improved, the weight vector is adjusted.
- The classification of patterns is defined on a symbolic level on the basis of the signs of the discriminating hyperplanes.

As an option in the case of non-convex (in particular non-singly connected) classes a clustering procedure decomposing the classes into subclasses can be applied. A standard minimum-squared-error algorithm with an initial partition depending on the sequence of presenting the patterns is used. Like hill-climbing algorithms in general, this approach guarantees local but not global optimisation. Another problem in finding an adequate clustering of a class (that means, a clustering which allows a linear discrimination from other classes resp. subclasses) is the data-based choice of an appropriate distance measure. In higher dimensions of the attribute space it is often quite impossible to find an appropriate scaling. The consequence is, that DIPOL generates more subclasses than

necessary to find a situation for linear separation of classes and subclasses, and this can result in overfitting of the training data.

### 2.2 The Decision Tree Algorithm CAL5

CAL5 induces a decision tree using a discretisation procedure which is especially suitable for continuous attributes. The goal of the splits at the nodes of the tree is to decrease the "impurity" of the learning subsets belonging to the nodes. CAL5 works with two impurity measures: a statistical measure and the information-theoretic entropy measure [5]. The accuracy rate and complexity of the resulting decision tree depend on two parameters - the confidence level and the dominance threshold. A coarse description of the procedure at each node of the tree is given in the following:

- The node is a leaf, if the probability of one class at the node is greater than the dominance threshold.
- If the node is no leaf, for each attribute
  - an automatic, adaptive discretisation is carried out on the basis of the prechosen confidence level and the dominance threshold and
  - the value of the impurity measure related to this discretisation is determined.

  The attribute with the least value of the impurity measure is chosen for the next split.

The algorithm stops, if each node is a leaf.
Tests on several real-world data sets show that the decision trees produced by CAL5 are usually quite compact in comparison with those generated by other algorithms, see [3], where also more details of the discretisation procedure and the used pruning method can be found.

### 2.3 The Hybrid Algorithm DIPOL-DT

This section describes the combination of the axis-parallel decision tree algorithm CAL5 and the piecewise linear classifier DIPOL. The hybrid algorithm chooses at each node the better of the DIPOL-split and the best axis-parallel CAL5-split. The procedure is fully deterministic and can be summarized as follows:

- **If all samples at the current node belong to the same class then STOP.**
- **Use DIPOL without any clustering to construct oblique hyperplanes (one in the case of two classes or more in the case of more than two classes).**
  **Use CAL5 with the entropy measure for evaluating the impurity to split the attribute space along one chosen attribute in two or more subspaces (construction of axis-parallel hyperplanes).**
  - **If no split is found: use DIPOL up to a prechosen number $n_c$ of subclasses of all classes and STOP with the best result.**

- **If one or more splits are found: evaluate the quality of the splits constructed by DIPOL and CAL5 using the entropy measure and decide in favour of the better splitting to form new branches of the tree.**

The hybrid algorithm is investigated empirically using artificial and real-world data sets. The examples confirm that the hybrid algorithm

- in general (i. e., when the underlying classification concept is defined by oblique hyperplanes) constructs compact trees in comparison with the axis-parallel decision tree algorithm CAL5 and
- substitutes the search for an adequate clustering of classes in cases, in which the classes do not allow a linear discrimination of each class from all others (when a linear discrimination of the classes is possible, DIPOL performs the classification).

Because the decision on the split at a node is made locally, the introduced algorithm (and all other known axis-parallel and oblique decision tree algorithms) does not generate the smallest possible tree, describing a given concept. But in particular real-world examples in higher than two-dimensional attribute spaces demonstrate that the algorithm DIPOL-DT generates significantly more compact classification concepts than DIPOL or CAL5 alone.
The performance of DIPOL-DT is compared to that of several other axis-parallel and oblique decision tree algorithms and will be presented by some artificial and real-world examples in the poster session.

## References

1. Breiman, L., Friedman, J., Olshen R. & Stone, C. (1984). Classification and Regression Trees. Wadsworth International Group.
2. Heath, D., Kasif, S. & Salzberg, S. (1993). Learning oblique decision trees. In Proceedings of the 13th International Joint Conference on Artificial Intelligence, pp. 1002-1007. Chambery, France, Morgan Kaufmann.
3. Michie, D., Spiegelhalter, D. J. & Taylor, C. C.(1994). Machine Learning, Neural and Statistical Classification. Ellis Horwood.
4. Murthy, S. K., Kasif, S. & Salzberg, S. (1994). A System for Induction of Oblique Decision Trees. Journal of Artificial Intelligence Research, 2, pp. 1-32.
5. Quinlan, J. R.(1986). Induction of Decision Trees. Machine Learning, 1, pp. 81-106.
6. Utgoff, P. E. (1988). Perceptron Trees: A Case Study in Hybrid Concept Representation. In Proceedings of the 6th National Conference of Artificial Intelligence, pp. 601-606.
7. Utgoff, P. E. & Brodley, C. E. (1991). Linear Machine Decision Trees. Technical Report 10, University of Massachusetts at Amherst.