FEATURE SPACE PARTITIONING BY NON-LINEAR AND FUZZY DECISION TREES

Andreas Ittner[a], Jens Zeidler[a], Rolf Rossius[a], Werner Dilger[a], Michael Schlosser[b]

[a] *Chemnitz University of Technology, Department of Computer Science, D-09107 Chemnitz, Phone: +49 371 531-1643, Fax: +49 371 531-1465, E-mail: {ait,jzei,ros,wdi}@informatik.tu-chemnitz.de*

[b] *FH Koblenz, Department of Electrical Engineering, Am Finkenherd 4, D-56075 Koblenz, Phone: +49 261 9528-187, Fax: +49 261 56953, E-mail: schlosser@koblenz.fh-rpl.de*

Abstract. This paper focuses on a unified sight in the field of non-linear feature space partitioning. We present two well-known approaches of growing decision trees from data and show that these methods have a lot in common regarding non-linearity. The aim of this paper is to clarify that the application of simple mathematical operations broadens the capabilities to split the feature space in a non-linear fashion.

Keywords: Non-linear and Fuzzy Decision Trees, Feature Space Partitioning

1 Introduction

In the research field of supervised learning from examples the accuracy of feature space partitioning is in some cases much more important than the simplicity of the splitting. One important method for more accurate separation of examples belonging to different classes is non-linear feature space partitioning. Especially in the field of growing decision trees from data non-linear partitioning broadens the capabilities of finding good splits in the feature space. One way to achieve the goal of non-linearity is to integrate a simple mathematical operation, namely multiplication. This can be done for instance within the process of generating new features from a set of given task-supplied primitive ones. Using these newly created 'non-linear' features allows for a non-linear partitioning of the original feature space. The studies in [3] and [2] showed that non-linear decision tree algorithms (NDT's) produce more accurate trees than their axis-parallel or oblique counterparts. On the other hand, multiplication is a quite important operation to define logical operators in fuzzy theory and its application on fuzzy decision trees (FDT's). In the area of FDT's [8] we can find non-linear partitionings too. Here the calculation with Fuzzy-AND and Fuzzy-OR in conjunction with defuzzification creates new non-linear geometrical forms of feature space partition (see [9]).

Section 2 of this paper is dedicated to non-linear feature space partitioning with NDT's. Section 3 deals with fundamentals of fuzzy-decision trees generation. In section 4 we try to develop a unified view of the partitioning problem as a synthesis of these two types of trees. Here we demonstrate the influence of membership functions and the fuzzy operators on the feature space partitioning. Section 5 summarizes the lessons learned from this unification and outlines an avenue for further work.

2 Non-Linear Decision Trees

The method of NDT, first introduced in [1], proceeds in two consecutive steps:

- augmentation of the feature space
- application of a decision tree algorithm.

The first step is done by building all possible pairwise products and squares of n given primitive numerical features, resulting in a set of $\frac{n^2+3n}{2}$ features which are considered as the axes of a new feature space. These features are represented by terms in equations describing hypersurfaces of the second degree. For example, in the two-dimensional case such an equation has the form

$$0 = ax_1^2 + 2bx_1x_2 + cx_2^2 + 2dx_1 + 2ex_2 + f. \quad (1)$$

Ellipses, circles, hyperbolas, etc. are described by equations of this type. In the m-dimensional case ($m > 2$) we get elliptoids, hyperboloids, paraboloids, etc.

In the second step of the NDT method a decision tree algorithm is applied to construct an oblique decision tree in the augmented feature space which is now of higher dimension.In our experiments we used OC1 (Oblique Classifier 1) [5]. This algorithm generated hyperplanes with an oblique orientation as a test of a linear combination of primitive and new created features at each internal node. In general, these hyperplanes corresponded to non-linear hypersurfaces in the original feature space of primitive features.

One of the data sets we used in our experiments [2] was an artificial set, called SPIRAL [Figure 1], which allows for an exemplary demonstration of the ability of the NDT method [Figure 2].

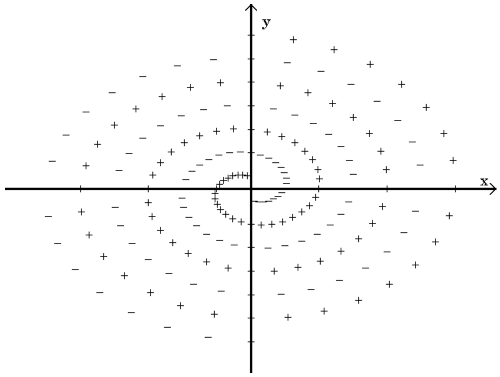

Figure 1: Spiral Data Set

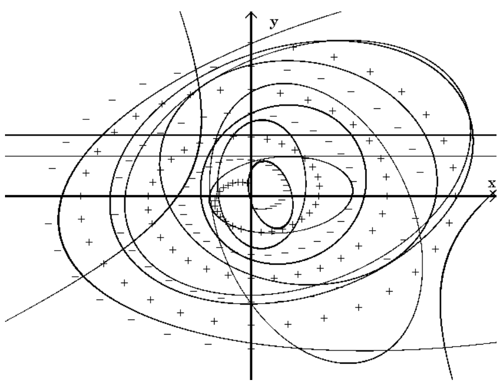

Figure 2: Non-Linear Partitioning

The source of power regarding non-linear feature space partitioning with NDT's was the utilization of dualism between a linear partitioning in a tricky constructed feature space of high dimension and the original space of the given features. In the next section we describe another approach of non-linear feature space partitioning which is based on FDT's.

3 Fuzzy Decision Trees

Classical crisp decision trees (i. e. axis-parallel, oblique, and non-linear decision trees) are widely applied to classification tasks. However, there is also a number of fuzzy decision tree solutions [4], [6], [7] and [8]. In the field of FDT's, the learning examples are labelled with membership grades. These grades represent the affiliations of examples to the classes. Fuzzy borders for discretisation of continuous-valued attributes [Figure 3] are used in almost all approaches mentioned above.

The resulting FDT consists of nodes and leafs. Every internal node corresponds to a test attribute [Figure 4]. In this example 'weight' and 'height' are the features. The leaf nodes are labeled by class membership values for all classes. These values represent the class ratio of examples with the attribute values according to the internal nodes from the root to the leaf node. Each branch is described by a membership-function according to the discretisation of the continuous-valued attribute.

To classify new unseen examples with an FDT we have to calculate all membership values. Here the usage of diagrams on the branches of the tree comes in. The membership values (along the way from the root to the leaves) are combined by each other and by the class membership at the each leaf. Often the class membership of unseen examples is equal to the sum of products for each class ($\times - \times - +$-method). In general, fuzzy operators (Fuzzy-AND and Fuzzy-OR) are used to calculate the class memberships of unseen examples.

In the following section we are going to deal with this particular part in detail. Furthermore we will show that NDT's and FDT's have a lot in common regarding non-linear feature space partitioning.

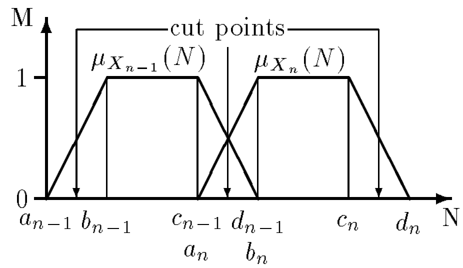

Figure 3: Trapezoidal membership functions

4 Non-Linear Feature Space Partitioning: The Source of Power

NDT's and FDT's share the use of the multiplication operation for the non-linear partitioning of the feature space. However, they differ from each other with respect to the way how this operation is applied. In case of NDT, multiplication is applied to the task-supplied primitive features to construct new ones. Multiplication, so to speak, has influence on the whole feature space. Compared with this in FDT's multiplication is used to create special fuzzy operations, like Fuzzy-AND and Fuzzy-OR.

The crux of a FDT is the partitioning of the whole feature space in axis-parallel basis cuboids (B-cuboids) based of the given membership functions D [Figure 5]. These B-cuboids contain all instances of a given training set, e. g. they correspond to the leafs of a generated FDT. The B-cuboids are bounded by the two top corner points ($a_n = c_{n-1}$ and $b_n = d_{n-1}$) of underlying trapezoidal membership functions [Figure 3]. The vectors $(C_{1,1},...,C_{2,2})$ of membership values of each class are calculated based on subsets of the training set. The class membership of an unseen test instance which is located in some B-cuboid is the vector of the corresponding B-cuboid.

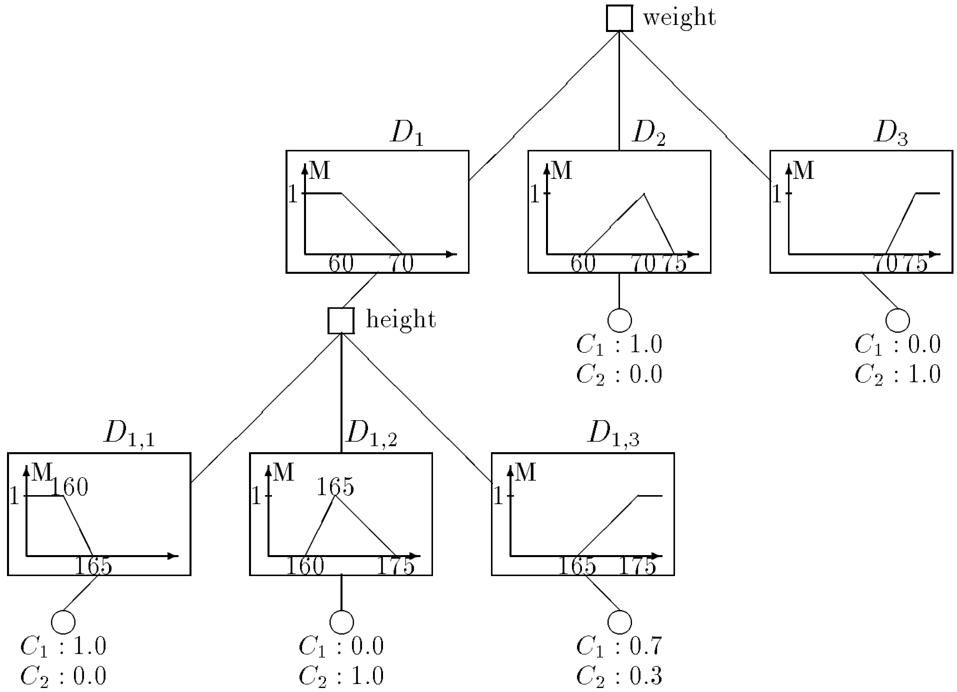

Figure 4: The FDT created from an example data set including membership functions [9]

Each B-cuboid influences its environment. This environment is determined by the legs of trapezoid membership functions. The environments induce the partitioning of the area between the B-cuboids in composite cuboids (C-cuboids) [Figure 5]. The class memberships (vectors C_I,...,C_{VII}) in the C-cuboids result as compositions of memberships in the influencing B-cuboids.

The membership in a C-cuboid is determined by the legs of two adjacent trapezoids, i. e. by linear functions. Therefore the composition of memberships along some path of an FDT is defined as a combination of such linear functions to non-linear ones that correspond to curves and surfaces of a higher degree. Thus it turns out that the membership functions for the elements in the C-cuboids have the same form as the hypersurfaces that are used to partition the whole feature space in the NDT method.

Figure 5 shows a possible partition of a two dimensional space. The FDT algorithm has cut the axis x_1 once, and each of the two vertical strips once again (one cut means a pair of top corner points of consecutive trapezoids and creates the two 'certain' areas and one area of uncertainty inbetween). In general there may be many strips D_1,..., D_n in the first level and more cuts of these strips, as well as a higher dimension with a deep hierarchy.

Examples of vectors of explicit polynomials are given in the following. Each one is defined over the C-cuboid described by the index of the vector. The polynomials over the remaining cuboids are similar.

$$C_I(x_2) = \alpha_1 x_2 + \alpha_2 \tag{2}$$

$$\alpha_1 = \frac{C_{1,2} - C_{1,1}}{d_{1,1} - c_{1,1}} \tag{3}$$

$$\alpha_2 = \frac{C_{1,1}d_{1,1} - C_{1,2}c_{1,1}}{d_{1,1} - c_{1,1}} \tag{4}$$

$$C_{VI}(x_1, x_2) = \alpha_1 x_1 x_2 + \alpha_2 x_1 + \alpha_3 x_2 + \alpha_4 \tag{5}$$

$$\alpha_1 = \frac{C_{1,1} - C_{1,2}}{(d_1 - c_1)(d_{1,1} - c_{1,1})} \tag{6}$$

$$\alpha_2 = \frac{C_{2,2}}{(d_1 - c_1)} + \frac{C_{1,2}c_{1,1} - C_{1,1}d_{1,1}}{(d_1 - c_1)(d_{1,1} - c_{1,1})} \tag{7}$$

$$\alpha_3 = \frac{(C_{1,2} - C_{1,1})d_1}{(d_1 - c_1)(d_{1,1} - c_{1,1})} \tag{8}$$

$$\alpha_4 = -\frac{C_{2,2}c_1}{(d_1 - c_1)} + \frac{(C_{1,1}d_{1,1} - C_{1,2}c_{1,1})d_1}{(d_1 - c_1)(d_{1,1} - c_{1,1})} \tag{9}$$

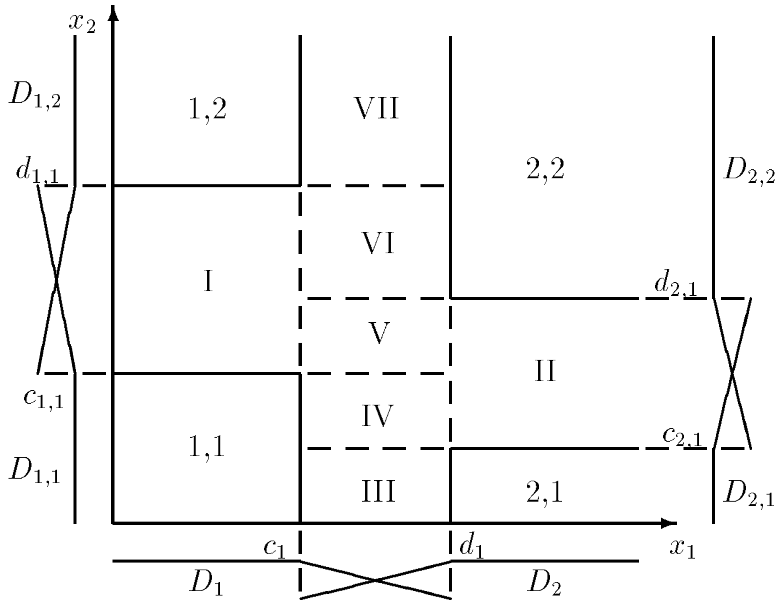

Figure 5: x_1-x_2-feature space with B-cuboids (1,1;...;2,2), C-cuboids (I...VII), and the trapezoid membership functions D_1,...,$D_{2,2}$

$$C_V(x_1, x_2) = \alpha_1 x_1 x_2 + \alpha_2 x_1 + \alpha_3 x_2 + \alpha_4 \quad (10)$$

$$\alpha_1 = \frac{C_{1,1} - C_{1,2}}{(d_1 - c_1)(d_{1,1} - c_{1,1})} + \frac{C_{2,2} - C_{2,1}}{(d_1 - c_1)(d_{2,1} - c_{2,1})} \quad (11)$$

$$\alpha_2 = \frac{C_{2,1} d_{2,1} - C_{2,2} c_{2,1}}{(d_1 - c_1)(d_{2,1} - c_{2,1})} + \frac{C_{1,1} d_{1,1} - C_{1,2} c_{1,1}}{(d_1 - c_1)(d_{1,1} - c_{1,1})} \quad (12)$$

$$\alpha_3 = \frac{(C_{2,1} - C_{2,2}) c_1}{(d_1 - c_1)(d_{2,1} - c_{2,1})} + \frac{(C_{1,2} - C_{1,1}) d_1}{(d_1 - c_1)(d_{1,1} - c_{1,1})} \quad (13)$$

$$\alpha_4 = \frac{(C_{1,1} d_{1,1} - C_{1,2} c_{1,1}) d_1}{(d_1 - c_1)(d_{1,1} - c_{1,1})} + \frac{(C_{2,2} c_{2,1} - C_{2,2} d_{2,1}) c_1}{(d_1 - c_1)(d_{2,1} - c_{2,1})} \quad (14)$$

5 Conclusion and Further Work

This paper has dealt with the question of a unified view in the field of non-linear feature space partitioning. We have described two approaches to reach this goal. On the one hand, non-linearity is achieved by the combination of task-supplied primitive features, on the other hand, fuzziness offers the opportunity of this kind of feature space splitting. As far as we know, up to now no attention has been paid on the area of feature space partitioning in conjunction with NDT- and FDT-classification in this way. The simple mathematical multiplication operation, the source of power, plays an important role in both the field of NDT's and in the area of FDT's. We plan to extend our work to develop a unified theory of non-linear partitioning of the feature space in general.

References

1. A. Ittner. Ermittlung von funktionalen Attributabhängigkeiten und deren Einfluß auf maschinelle Lernverfahren. Master's thesis, Dept. of Computer Science, Chemnitz University of Technology, Germany, 1995. Only in German available.

2. A. Ittner and M. Schlosser. Discovery of relevant new features by generating non-linear decision trees. In E. Simoudis, J. Han, and U. Fayyad, editors, *Proc. of 2 nd International Conference on Knowledge Discovery and Data Mining*, pages 108–113. AAAI Press, Menlo Park, CA, Portland, Oregon, USA, 1996. http://www.tu-chemnitz.de/˜ait/publications/kdd96.ps.gz.

3. A. Ittner and M. Schlosser. Non-linear decision trees - NDT. In L. Saitta, editor, *Proc. of 13 th International Machine Learning Conference*, pages 252–257. Morgan Kaufmann, San Francisco, CA, Bari, Italy, 1996. http://www.tu-chemnitz.de/˜ait/publications/icml96.ps.gz.

4. C. Z. Janikow. Fuzzy processing in decision trees. In *Proceedings of International Symposium on Artifical Intelligence*, pages 360–367, 1993.

5. S. Murthy, S. Kasif, S. Salzberg, and R. Beigel. OC1: Randomized induction of oblique decision trees. In *Proc. of the 11 th Nat. Conf. on*

AI AAAI-93, pages 322–327, Washington, D.C., 1993.

6. M. Umano, H. Okamoto, I. Hatono, H. Tamura, F. Kawachi, S. Umedzu, and J. Kinoshita. Fuzzy decision trees by fuzzy ID3 algorithm and its application to diagnosis systems. In *Proc. of* 3 rd *IEEE International Conference on Fuzzy Systems*, pages 2113–2118, Orlando, FL, 1994.

7. X. Wu and P. Måhlén. Fuzzy interpretation of induction results. In U. M. Fayyad and R. Uthurusamy, editors, *Proc. of 1* st *International Conference on Knowledge Discovery and Data Mining*, pages 325–330, Montreal, Quebec, Canada, 1995.

8. J. Zeidler and M. Schlosser. Fuzzy handling of continuous-valued attributes in decision trees. In Y. Kodratoff, G. Nakhaeizadeh, and Ch. Taylor, editors, *Proc. of MLNet Familiarization Workshop: Statistics, Machine Learning and Knowledge Discovery in Databases*, pages 41–46, Heraklion, Crete/Greece, 1995. http://www.tu-chemnitz.de/~jzei/VEROEFF/ecws3.ps.

9. J. Zeidler and M. Schlosser. Continuous-valued attributes in fuzzy decision trees. In *Proc. of* 6 th *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 395–400, Granada, Spain, 1996. http://www.tu-chemnitz.de/~jzei/VEROEFF/ipmu.ps.