Generating Decision Trees and Membership Functions by Fuzzy Clustering

Thomas A. Runkler
Siemens AG
Corporate Technology
81730 Munich, Germany
Phone: ++49 89 636 45372
Fax: ++49 89 636 45456
Email: Thomas.Runkler@mchp.siemens.de

Shounak Roychowdhury
Oracle Corporation
500 Oracle Parkway
Redwood Shores, CA 94065, U.S.A.
Phone: ++1 650 506 1720
Fax: ++1 650 506 7418
Email: sroychow@us.oracle.com

ABSTRACT: There are various algorithms to generate decision trees (DTs) from data. ID3 and CART can be used to generate DTs for real data with predefined classes. Fuzzy ID3 and fuzzy CART are fuzzifications of these algorithms which need predefined fuzzy classes. We present a new method to generate crsip or fuzzy DTs from data based on clustering. Our "decision tree clustering" (DTC) automatically determines not only the structure of the decision tree, but also the fuzzy classes and class borders from the data set.

I. GENERATION OF DECISION TREES

Induction on decision trees (ID3) [10] is a method for generating decision trees from discrete (object) data sets O or real valued data sets $X \in \mathbb{R}^p$ with predefined classes. From all possible permutations of possible decision trees ID3 determines an optimal tree maximizing the information gain obtained for each branch when traversing from the root to the leaves. The classification and regression tree algorithm (CART) [4] generates decision trees for real valued data $X \in \mathbb{R}^p$ assigning each component X_l, $l = 1, \ldots, p$, of X two crisp binary classes L_l and H_l separated by an interval limit s_l, so that $L_l = (-\infty, s_l)$ and $H_l = [s_l, \infty)$.

There are various approaches to fuzzify the ID3 and CART algorithms. Maher and St. Clair [9] used ID3 to construct decision trees and then fuzzified the labels for recall using uncertain reasoning (UR), hence this algorithm is called UR–ID3. Janikow [8] extended the ID3 approach by allowing the predefined classes for real valued $X_l \in \mathbb{R}$, $l = 1, \ldots, p$, to be fuzzy (fuzzy ID3, FID3). In FID3 the user defines membership functions $\mu_i^{(l)} : \mathbb{R} \to [0,1]$, $i \in \{2, 3, \ldots\}$, for each component X_l. Each membership function $\mu_i^{(l)}$ can serve as an edge label of the fuzzy decision tree (FDT). As in ID3, FID3 generates an optimal decision tree by maximizing information gains. Hayashi et al. [5] parametrized FID3 by using the family of Schweizer/Sklar t-norms and t-conorms [11]. Applications of FID3 are reported in [1]. Jang [7] suggests a fuzzification of the crisp CART results, where the decision tree obtained by CART is used to define initial rules which are subsequently refined using his ANFIS algorithm [6].

The main drawback of FID3 is that the membership functions have to be specified by the user. For high dimensions p or ill defined data this is a difficult task requiring intensive involvement of experts. On the other hand, the necessary information to obtain reasonable membership functions might be contained in the data set X itself. This information is used in a large quantity of fuzzy modeling algorithms to extract membership functions from data. Therefore, our goal here is to use the data set X to extract automatically both, an FDT and the corresponding membership functions. Notice that we cannot use CART for this task, because CART only generates crisp classes, and even Jang's "fuzzy CART" only uses the crisp CART algorithm to generate the decision tree. Here, we are interested in one integrated algorithm for the extraction of a fuzzy decision tree and its membership functions.

II. MODELING BY CLUSTERING

A popular method to extract membership functions from data is clustering. Membership functions characterizing fuzzy partitions can be obtained using the fuzzy c–means (FCM) model [3] defined as the following problem:

Given the data set X and a fuzziness parameter $m \in (1, \infty)$, minimize the objective function

$$J_{\mathrm{FCM}}(U, V; X) = \sum_{k=1}^{n} \sum_{i=1}^{c} u_{ik}^{m} |x_k - v_i|^2, \tag{1}$$

where $U = \{u_{ik}\}$, $V = \{v_i\}$, $u_{ik} \in [0,1]$ is the membership of x_k in the i^{th} cluster, $i = 1, \ldots, c$, $k = 1, \ldots, n$, with $\sum_{i=1}^{c} u_{ik} = 1$, for all $k = 1, \ldots, n$, and v_i is the center of the i^{th} cluster, $i = 1, \ldots, c$. The FCM model can be optimized by alternating optimization (AO) through the necessary conditions for extrema of J_{FCM}. In FCM–AO, memberships and cluster centers are alternatingly updated as

$$u_{ik} = 1 \Bigg/ \sum_{j=1}^{c} \left(\frac{|x_k - v_i|}{|x_k - v_j|} \right)^{\frac{2}{m-1}} \quad \text{and} \tag{2}$$

$$v_i = \frac{\sum_{k=1}^{n} u_{ik}^{m} x_k}{\sum_{k=1}^{n} u_{ik}^{m}}, \tag{3}$$

until subsequent estimations V and V^* of the cluster centers satisfy $\max_{i=1,\ldots,c} \max_{l=1,\ldots,p} (v_i^{(l)} - v_i^{*(l)}) < v_{th}$, where v_{th} is a threshold parameter.

Continuous membership functions $\mu_i^{(l)} : \mathbb{R} \to [0,1]$, $i = 1, \ldots, c$, $l = 1, \ldots, p$, can be obtained by projection and subsequent interpolation or approximation of the u_{ik}, or simply by inserting the projections $v_i^{(l)}$ of the cluster centers v_i into

$$\mu_i^{(l)}(x^{(l)}) = 1 \Bigg/ \sum_{j=1}^{c} \left(\frac{|x^{(l)} - v_i^{(l)}|}{|x^{(l)} - v_j^{(l)}|} \right)^{\frac{2}{m-1}}. \tag{4}$$

III. ID3 WITH CLUSTERING

In order to design an integrated algorithm extracting a decision tree and the corresponding membership functions we combine the extraction of membership functions using clustering with the determination of a fuzzy decision tree based on (F)ID3. This algorithm called *"decision tree clustering" (DTC)* works as follows:

At the beginning of the algorithm, the decision tree is empty, we only have one (root) node N. To decide, which variable should be used to proceed to the children of N, each dimension $l = 1, \ldots, p$ of the data set X is clustered with various cluster numbers $c = 2, 3, \ldots, c_{\max}$ leading to the partitions $U_c^{(l)}$ and prototypes $U_c^{(l)}$. For each partition $U_c^{(l)}$ we compute the partition coefficient [3]

$$\mathrm{PC}(U_c^{(l)}) = \frac{1}{n} \sum_{k=1}^{n} \sum_{i=1}^{c} \left(u_{ikc}^{(l)} \right)^2, \tag{5}$$

which quantifies how good the achieved partitions are. The best partition is obtained for the maximum $\mathrm{PC}(U_{c^*}^{(l^*)}) = \max_{c,l} \{\mathrm{PC}(U_c^{(l)})\}$. The corresponding dimension l^* is used to separate the data set, and the cluster number c^* indicates the number of subclasses that have to be generated. These subclasses are inserted into the decision tree by constructing c^* arcs at N with the labels $x^{(l^*)} < b_1^{(l^*)}$, $b_1^{(l^*)} \le x^{(l^*)} < b_2^{(l^*)}$, ..., $b_{c^*-2}^{(l^*)} \le x^{(l^*)} < b_{c^*-1}^{(l^*)}$, $b_{c^*-1}^{(l^*)} \le x^{(l^*)}$. The boundaries $b_i^{(l^*)}$, $i = 1, \ldots, c^* - 1$, are determined, so that the adjacent membership functions defined by (4) are equal. Without loss of generality assume that the cluster centers are sorted, so that $v_1^{(l^*)} < v_2^{(l^*)} < \ldots < v_{c^*}^{(l^*)}$. Then the borders $b_i^{(l^*)}$, $i = 1, \ldots, c^* - 1$, can be determined using

$$\mu_i^{(l^*)}(b_i^{(l^*)}) = \mu_{i+1}^{(l^*)}(b_i^{(l^*)}) \tag{6}$$

$$\Rightarrow \quad 1 \Bigg/ \sum_{j=1}^{c^*} \left| \frac{b_i^{(l^*)} - v_i^{(l^*)}}{b_i^{(l^*)} - v_j^{(l^*)}} \right|^{\frac{2}{m-1}} = 1 \Bigg/ \sum_{j=1}^{c^*} \left| \frac{b_i^{(l^*)} - v_{i+1}^{(l^*)}}{b_i^{(l^*)} - v_j^{(l^*)}} \right|^{\frac{2}{m-1}} \tag{7}$$

$$\Rightarrow \quad |b_i^{(l^*)} - v_i^{(l^*)}| = |b_i^{(l^*)} - v_{i+1}^{(l^*)}| \tag{8}$$

$$\Rightarrow \quad b_i^{(l^*)} = \frac{v_i^{(l^*)} + v_{i+1}^{(l^*)}}{2}. \tag{9}$$

At the next level of the decision tree we consider the data sets X_i for each child $i = 1, \ldots, c^*$ of N. The data sets X_i contain only $p - 1$ dimensional data, namely the p original ones except dimension l^*. Each X_i contains the

subset of this $p-1$ dimensional projection of the data set X that fulfills the condition label $b_{i-1}^{(l^*)} \leq x^{(l^*)} < b_i^{(l^*)}$, where $b_0^{(l^*)} = -\infty$ and $b_{c^*}^{(l^*)} = \infty$. The whole process is repeated recursively for each child, until there is no dimension or no data left.

Notice that this algorithm divides each one–dimensional projection of the data set into at least two subclasses, because the number of clusters is always $c > 1$. Some of the data components, however, might not possess any cluster structure at all, so these components can be ignored in order to get a smaller decision tree. This can be done by testing the data set for cluster structure before each clustering, i.e. it is checked, if there are at least two clusters contained. We used the PC to decide which cluster number gives the best partition. To check, if there is no cluster structure, we could formally interpret this as *only one* cluster. The corresponding PC is then computed for $c = 1$ and $u_{1k}^{(l)} = 1$ for all $i = 1, \ldots, n$:

$$\mathrm{PC}(U_1^{(l)}) = \frac{1}{n} \sum_{k=1}^{n} 1 = \frac{1}{n} \cdot n = 1. \tag{10}$$

For every fuzzy partiton U, however, we always have $\mathrm{PC}(U) < 1$ for any $c > 1$. So, this approach would always prefer $c^* = 1$ with $\mathrm{PC}(U) = 1$, which makes this method impracticable. We therefore use an alternative method here to test for $c^* > 1$: Let $x_{\min}^{(l)}$ and $x_{\max}^{(l)}$ denote the domain limits of each variable $x^{(l)}$ in the whole p dimensional data set X, and let $\xi_{\min}^{(l)}$ and $\xi_{\max}^{(l)}$ denote the domain limits of the variable $x^{(l)}$ in the actually considered q dimensional subset of X, $q \in \{1, \ldots, p\}$. Then the domain ratio

$$r^{(l)} = \frac{\xi_{\max}^{(l)} - \xi_{\min}^{(l)}}{x_{\max}^{(l)} - x_{\min}^{(l)}} \tag{11}$$

quantifies, which percentage of the whole domain is covered by the actually considered subset. If $r^{(l)} < r_{th}$, then the actual subset is considered small enough to be represented by one cluster only, hence $c^* = 1$, where $r_{th} \in [0,1]$ is a threshold parameter. If $r^{(l)} \geq r_{th}$, then clustering is performed for $c = 2, \ldots, c_{\max}$, and the optimum cluster number c^* is determined from the result with the maximum partition coefficient.

IV. EXPERIMENTS

We applied our algorithm to the two dimensional data sets from Janikow's paper [8] $X = \{ \binom{8}{58}, \binom{8}{54}, \binom{9}{53}, \binom{10}{48}, \binom{16}{50}, \binom{10}{45}, \binom{12}{12}, \binom{14}{16}, \binom{16}{16}, \binom{18}{15}, \binom{8}{8}, \binom{8}{10}, \binom{9}{12}, \binom{10}{10}, \binom{10}{13}, \binom{12}{12}, \binom{14}{16}, \binom{16}{16}, \binom{18}{15}, \binom{30}{30}, \binom{32}{28}, \binom{34}{35}, \binom{36}{37}, \binom{36}{32}, \binom{38}{28}, \binom{52}{56}, \binom{52}{52}, \binom{52}{53}, \binom{49}{53} \}$ which is plotted in Figure 1. X contains four clearly visible clusters, which can be separated into a cluster pair on the left and two single clusters in the middle and on the right by vertical lines at, e.g., $x^{(1)} = 25$ and $x^{(1)} = 45$. The cluster pair on the left can the be separated by a horizontal line at, e.g., $x^{(2)} = 30$.

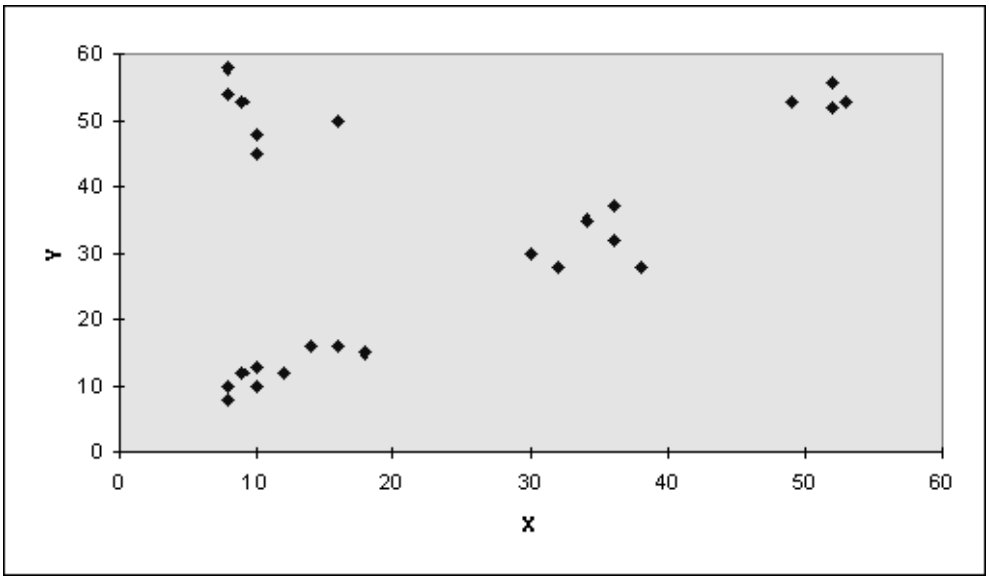

Figure 1. The data set X_1 containing four clusters.

We ran our DTC algorithm with fuzzy c–means clustering and the parameters $m = 2$, $c_{\max} = 4$, $v_{th} = 10^{-10}$, and $r_{th} = 0.3$, and obtained the decision tree shown in Figure 2. The decision tree exactly corresponds to the separation we described above. The positions of the borderlines for the $x^{(1)}$ and $x^{(2)}$ axes, however, are slightly different, because they are automatically determined from the clustering results: $x^{(1)} = 22.31$, $x^{(1)} = 41.19$, and $x^{(2)} = 31.32$. The corresponding *crisp* partition, however, is exactly the same as the partition with the visually obtained borders from above, i.e. the points $x_k \in X$ belong to the same (crip) classes. Notice that there are alternative ways to partition the data set X, e.g. by $x^{(1)} = 25$, $x^{(2)} = 30$, and $x^{(2)} = 45$. These results, however, are suboptimal with respect to cluster separability.

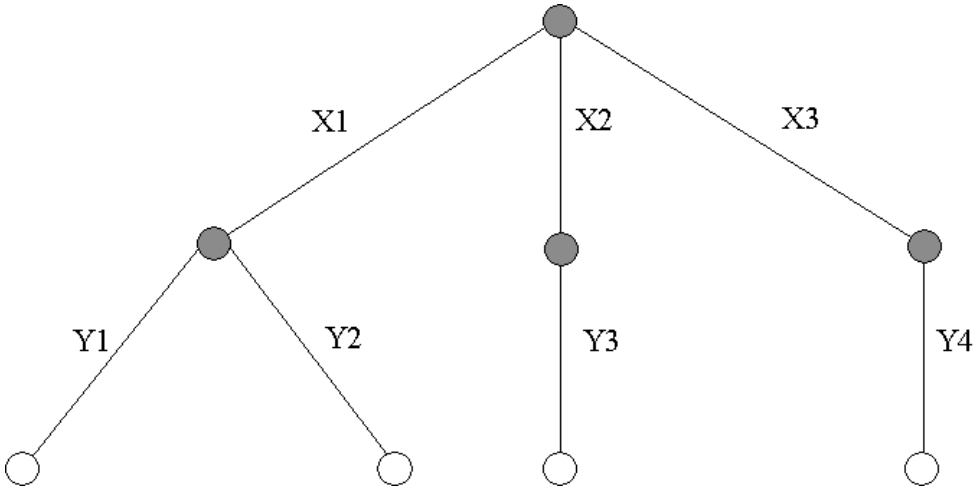

Figure 2. The decision tree obtained for the data set X_1. The (defuzzified) tree labels are X1 = ($8.00 \leq$ X $\leq$ 22.31), X2 = ($22.31 \leq$ X $\leq$ 41.19), X3 = ($41.19 \leq$ X $\leq$ 53.00), Y1 = ($8.00 \leq$ Y $\leq$ 31.32), Y2 = ($31.32 \leq$ Y $\leq$ 58.00), Y3 = ($28.00 \leq$ Y $\leq$ 37.00), and Y4 = ($52.00 \leq$ Y $\leq$ 56.00).

V. SUMMARY

Decision tree clustering (DTC) automatically extracts a decision tree plus the corresponding (crisp or fuzzy) classes from data in a unified approach. The decision tree is built by subsequent clustering of single dimensions, and the choice of the winner separation is based on cluster validity. We have presented an instance of this algorithm using the fuzzy c–means model (FCM) for clustering and the partition coefficient (PC) to determine the best separations. This DTC instance produced good results for Janikow's data set. Since PC is used as the validity measure, the results are optimal with respect to cluster separability. Other optimality conditions can be incorporated by choosing other validity measures. Moreover, also clustering models other than FCM can be used for generating decision trees. The use of the hard c-means model (HCM) [2] instead of FCM, for example, leads to crisp decision trees. DTC with HCM can therefore be regarded as an alternative to CART.

REFERENCES

[1] G. Adorni, D. Bianchi, and S. Cagnoni. Ham quality control by means of fuzzy decision trees: a case study. In *Proc. IEEE International Conference on Fuzzy Systems*, pages 1583–1588, Anchorage, May 1998.

[2] G. H. Ball and D. J. Hall. Isodata, an iterative method of multivariate analysis and pattern classification. In *Proc. IFIPS Congress*, 1965.

[3] J. C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms.* Plenum Press, New York, 1981.

[4] L. Breiman, J. H. Friedman, R. A. Olsen, and C. J. Stone. *Classification and Regression Trees.* Wadsworth, Belmont, 1984.

[5] I. Hayashi, T. Maeda, A. Bastian, and L. C. Jain. Generation of fuzzy decision trees by fuzzy ID3 with adjusting mechanism of and/or operators. In *Proc. IEEE International Conference on Fuzzy Systems*, pages 681–685, Anchorage, May 1998.

[6] J. S. R. Jang. ANFIS: Adaptive–network-based fuzzy inference system. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(3):665–685, May 1993.

[7] J. S. R. Jang. Structure determination in fuzzy modeling: A fuzzy CART approach. In *Proc. IEEE International Conference on Fuzzy Systems*, pages 480–485, Orlando, June 1994.

[8] C. Z. Janikow. Fuzzy decision trees: Issues and methods. *IEEE Transactions on Systems, Man, and Cybernetics*, 28(1):1–14, 1998.

[9] P. E. Maher and D. St. Clair. Uncertain reasoning in an ID3 machine learning framework. In *Proc. IEEE International Conference on Fuzzy Systems*, pages 7–12, San Francisco, March 1993.

[10] J. R. Quinlan. Induction on decision trees. *Machine Learning*, 11:81–106, 1986.

[11] B. Schweizer and A. Sklar. Associative functions and statistical triangle inequalities. *Publicationes Mathematicae Debrecen*, 8:169–186, 1961.