bETAs
New Simple Fuzzy Splitting Criteria for Inductive Learning*

G. Ramos-Jiménez[1], R. Morales-Bueno[1], F. Triguero-Ruiz[1], A. Villalba-Soria[2]

[1] Dpto. Lenguajes y Ciencias de la Computación, Universidad de Málaga
Bulevar Louis Pasteur 35, 29071 Málaga, Spain
{ramos, morales}@lcc.uma.es
[2] Student of E.T.S. Ingeniería Informática,

Abstract. A family (*b*eta) of new attribute selection criteria for TDIDT (Top Down Induction Decision Trees) algorithms is introduced. These splitting criteria are based upon the majority class (*b*est class). They are permissible (following the definition introduced by Michael Kearns and Yishay Mansour in 1999), allow us to work with experiences with fuzzy classes, easy to compute (no algorithm appears in their definition), and produce good empirical results.
Keywords: fuzzy splitting criterion, fuzzy inference systems, TDIDT, machine learning.

1 Introduction

Inductive learning is a branch of Artificial Intelligence of which the family of TDIDT algorithms is well-known [2][13]. From a set of experiences, these algorithms build a decision tree [9]. ID3 [15][16] is the best-known algorithm of this family. Developed by J.R. Quinlan in 1979, this algorithm builds decision trees by using Entropy [21] as the splitting criterion. Other splitting criteria, crisp as well as fuzzy, have been defined [3][8][11][12][16]. In general, every TDIDT algorithm [1][5][17][20][22][23] needs a splitting criterion.

We define new splitting criteria which fulfil three objectives:

1) That all are permissible following Michael Kearns and Yishay Mansour's [10] definition, i.e., that for two crisp classes each are: symmetric, concave, take the value one at the mid-point and value zero at points zero and one.

* This work has been partially supported by FRESCO project, number PB98-0937-C04, of CICYT Spain.

2) That they can be applied to experiences with fuzzy classes.

3) That their computation is simple and free of logarithms in their formulation.

The first ensures a certain goodness-of-fit at the theoretical level [10] if boosting techniques are utilized [6].
The second is very useful as there are applications in which the attributes are crisp but the classes are fuzzy. For example, there are medical problems where we have crisp results proceeding from clinical analysis, but where the classification of the patient or disease is done by the doctor in a fuzzy way.
The third is useful where the speed and reliability of the computation is critical, e.g., in real-time learning applications or those that handle huge data sets (terabytes).
In this paper we define four new splitting criteria *b1*, *b2*, *b3*, and *b4*, which fulfil the proposed criteria. All splitting criteria are based on the majority class (the *b*est class).
In section 2 we develop the concepts that we will use. In section 3 we define the new splitting criteria in which it is made clear that they are easy to compute. The goodness of the splitting criteria is shown in section 4 by means of experimental results. Finally, the conclusions are presented.

2 Notation And Concepts

We assume knowledge of TDIDT algorithms and the splitting criteria cited above. Only the concepts necessary to define the new splitting criteria are introduced.
The following notation is in line with our previous works [18][19]:

A *problem* with a attributes and k class is a vector $(\underline{m}, k) \in \mathsf{N}^a \times \mathsf{N}$, where $\underline{m} = (m_1, ..., m_a)$.
The domain of each attribute X_i is called $D_i = \{1, 2, ..., m_i\}$, $i = 1, ..., a$.
In addition, to avoid partial functions, an attribute X_0 is included with domain $D_0 = \varnothing$.
The attribute set is denoted by X: $X = \{X_0, X_1, ..., X_a\}$.
A special attribute with k values called *class* is denoted by C, and its domain D is codified $D = \{1, 2, ..., k\}$.

Example 1. Let us consider two attributes: height and weight, and the class "structure" with three values (proportioned, well-proportioned, and very well-proportioned). The attributes can be considered as discrete with four height intervals and five weight intervals. This problem is defined by:
$(\underline{m}, 3) \in \mathsf{N}^2 \times \mathsf{N}$, $\underline{m} = (4, 5)$,
$X = \{X_0, X_1 = height, X_2 = weight\}$,
$D_0 = \varnothing$
$D_1 = \{1, 2, 3, 4\}$
$D_2 = \{1, 2, 3, 4, 5\}$
And for C=structure, the domain is $D = \{1, 2, 3\}$

Remark 1. This representation is mentioned in [17]. The author, J. Ross Quinlan, states: "This transformation does not lose any of the functionality of ordinal attributes, but does make the resulting classifier more difficult to understand." However, it is possible to do in the output the reverse translation, in order to understand the results.

An observation for a problem $(\underline{m}, k)$ with $\underline{m} = (m_1, ..., m_a)$ is a vector with values of the a attributes $obs = (v_1, v_2, ..., v_a)$ with $v_i \in D_i$ $i=1,...,a$.
We consider the fuzzy set C_j $j=1,...,k$. Thus, the observation obs belongs to each C_j with different degrees: $\mu_{C_1}(obs) = a_1, \ldots, \mu_{C_k}(obs) = a_k$.
We need a simplified notation where these degrees are components of an amplified vector of the observation, called experience. Additionally, we will write $C_j(obs)$ instead of $\mu_{C_j}(obs)$. These notations are condensed in the following:

Let us consider a problem $(\underline{m}, k) \in \mathbb{N}^a \times \mathbb{N}$, with a attributes whose domains are D_1, D_2, … , D_a.
We define the universe of experiences $U_E = D_1 \times D_2 \times \ldots \times D_a \times [0,1]^k$. An experience e is an element of U_E, that is, a vector with $a+k$ components: $e = (X_1(e), X_2(e), ..., X_a(e), C_1(e), C_2(e), ..., C_k(e)) \in U_E$, where $X_i(e)$ is the value of the i attribute in the e experience and $C_j(e)$ is the membership degree of the e experience to the j class.
We will work with finite sequences of experiences $E = \{e_1, e_2, ..., e_N\}$, where some elements could be repeated.
The set of all finite sequences of experiences is represented by $\mathbb{E}$.

Example 2. Let us consider the problem described in example 1.
An experience would be, for example, $e = (3, 2, 0.3, 0.8, 0.1)$, $X_1(e)$=3 (third interval of height), $X_2(e)$=3 (second interval of weight), $C_1(e)$=0.3 (proportioned), $C_2(e)$=0.8 (well-proportioned), $C_3(e)$=0.1 (very well-proportioned).
An example of E, that will be used later, is as follows: $E = \{ e_1, ..., e_{20} \}$ with:

$e_1 = (4, 4, 0.4, 0.1, 0.7)$ $\quad$ $e_2 = (4, 5, 0.1, 0.1, 0.9)$
$e_3 = (4, 5, 0, 0.1, 0.8)$ $\quad$ $e_4 = (4, 4, 0.4, 0.1, 0.7)$
$e_5 = (3, 3, 0.2, 0.5, 0.8)$ $\quad$ $e_6 = (3, 3, 0.2, 0.9, 0.4)$
$e_7 = (3, 3, 0.1, 0.7, 0.3)$ $\quad$ $e_8 = (3, 5, 1, 0.1, 0)$
$e_9 = (3, 1, 0.8, 0.2, 0.1)$ $\quad$ $e_{10} = (2, 1, 0.4, 0.6, 0.1)$
$e_{11} = (2, 4, 0.9, 0, 0.3)$ $\quad$ $e_{12} = (2, 2, 0.6, 0.6, 0.2)$
$e_{13} = (2, 2, 0.4, 0.7, 0.1)$ $\quad$ $e_{14} = (1, 5, 1, 0.1, 0.1)$
$e_{15} = (1, 5, 0.9, 0.2, 0.2)$ $\quad$ $e_{16} = (1, 1, 0.4, 0.6, 0)$
$e_{17} = (1, 1, 0.2, 0.5, 0.1)$ $\quad$ $e_{18} = (1, 5, 0.8, 0.1, 0.1)$
$e_{19} = (1, 1, 0.5, 0.6, 0.2)$ $\quad$ $e_{20} = (1, 3, 1, 0.1, 0)$

where, for example, $e_1 = e_4$.

A *splitting criterion* is a function: $criterion : E \rightarrow R$. A real value is assigned to a sequence of experiences. Usually, the splitting criteria are defined normalized to the $[0,1]$ interval, i.e., $criterion: E \rightarrow [0,1]$.
In this paper we consider normalized splitting criteria. This decision is not critical because each splitting criterion selects the same attribute in both versions (normalized or otherwise).

3 The *b* Splitting Criteria

Let $(\underline{m}, k)$ be a problem. Let $E = \{e_1, e_2, ..., e_N\}$ be a finite sequence of experiences for this problem.
We define the *M* function as:

$$M : U_E \rightarrow D \qquad (1)$$

$$M(e) = min\{ j \in D \mid C_j(e) = max\{ C_1(e), C_2(e), ..., C_k(e) \} \}$$

and we define the *S* function as:

$$S : U_E \times D \rightarrow \{0,1\} \qquad (2)$$

$$S(e,j) = 1 \quad if \quad M(e) = j$$

$$S(e,j) = 0 \quad if \quad M(e) \neq j$$

For $j=1, \ldots, k$ let:

$$r_j = \frac{\sum_{i=1}^{N} S(e_i, j)}{N} \qquad (3)$$

be the rate of experiences in *E* where the *j* class has the greatest degree of membership.

Example 3. Let us consider the problem and the sequences of experiences *E* described in example 2.
For $e_1 = (4, 4, 0.4, 0.1, 0.7)$ we have:
$M(e_1) = 3$ and $S(e_1, 1) = 0$, $S(e_1, 2) = 0$, $S(e_1, 3) = 1$.
For $E = \{e_1, \ldots, e_{20}\}$ we have:
$r_1 = 8/20 = 0.4$, $r_2 = 7/20 = 0.35$, $r_3 = 5/20 = 0.25$.

Let $r_{1^o}, r_{2^o}, r_{3^o}, \ldots, r_{k^o}$ be the r_j values in decreasing order.
We define:

$$r_{max} = max\{ r_1, r_2, \ldots, r_k \} \qquad (r_{max} = r_{1^o}) \qquad (4)$$

The splitting criterion *b1* is defined as follows:

$$b1 : E \rightarrow [0,1] \qquad (5)$$

$$b1(E) = 1 - r_{max}$$

The second splitting criterion *b2* is defined as follows:

$$b2 : E \rightarrow [0,1] \qquad (6)$$

$$b2(E) = 1 - sumax$$

$$\text{where} \quad sumax = r_{max} - r_{max}^2 + \sum (r_i)^2$$

Another splitting criterion *b3* is defined as follows:

$$b3 : E \rightarrow [0,1] \qquad (7)$$

$$b3(E) = 1 - supot$$

$$\text{where} \quad supot = \sum_{n=1}^{k} (r_{n^o})^n$$

Finally, we define the splitting criterion *b4*:

$$b4 : E \rightarrow [0,1] \qquad (8)$$

$$b4(E) = 1 - sum$$

$$\text{where} \quad sum = \sum (r_i)^2$$

Example 4. Let us consider the problem and the sequences of experiences *E* described in example 2.

For $E = \{ e_1 , \ldots , e_{20} \}$ we have:

$b1(E) = 1 - r_{max} = 1 - 0.4 = 0.6$

$b2(E) = 1 - sumax = 1 - (0.4 + (0.35)^2 + (0.25)^2) = 1 - 0.585 = 0.415$

$b3(E) = 1 - supot = 1 - (0.4 + (0.35)^2 + (0.25)^3) = 1 - 0.538125 = 0.461875$

$b4(E) = 1 - sum = 1 - ((0.4)^2 + (0.35)^2 + (0.25)^2) = 1 - 0.345 = 0.655$

If all experiences in *E* have the greatest degree of membership in the same class then the values of these four splitting criteria are zero (we consider that *E* is totally ordered).

b1 indicates the fraction of experiences erroneously classified if the prediction of the majority class is considered.

b2, *b3*, and *b4* take into account the prediction of the majority class and the dispersion of erroneous experiences in the remainder class. There is more order when the sizes of the erroneous classes are more unequal.

b2 and *b3* are weighted criteria and *b4* is not.
The difference between *b2* and *b3* is that in *b2* all erroneous classes have the same weight, and in *b3* these weights are decreasing.

Remark 2. The splitting criteria *b2*, *b3*, and *b4* for two classes coincide -- if these are crisp then they are equal to Gini's criterion -- from which it can be deduced that their quadratic roots coincide with Kearns and Mansour's criterion. This criterion has the least maximum theoretical bound for learning by boosting with TDIDT algorithms [10].

These splitting criteria are used to obtain a decision tree. These form our basis to define the *gradient* function, that is, a function to select the attribute for each node [18][19].

Given a *criterion*, $criterion' : X \times E \rightarrow [0,1]$ is defined as follows:

$$criterion'(X_i, E) = \left(\sum_{j=1}^{m_i} criterion(E_j) \right) / N \quad \text{where}$$

$E_j = \{ e \in E \mid X_i(e) = j \}$ and $N = |E|$

Then, the *gradient function*, $\Delta : X \times E \rightarrow [0,1]$ is defined by:

$$\Delta(X_i, E) = criterion(E) - criterion'(X_i, E)$$

By considering the criteria *b1*, *b2*, *b3* or *b4*, the modified criteria are, respectively: *b1′*, *b2′*, *b3′* or *b4′*.
The selected attribute is the attribute that maximizes the gradient function (for the considered criterion).
All the *b*eta criteria are easy to compute, especially the first one. Thus, these can be useful when the time of learning is critical (learning in real-time) or when the data set is very big (terabytes).
The next section shows that these criteria also produce good results.

4 Experimental Results

We have applied the TDIDT algorithm, without pruning, by using five different splitting criteria: classical Entropy (ID3) and the four new splitting criteria. The experiment was carried out without pruning so that the only factor responsible for the different results was the different splitting criteria used. The application is carried out on the four standard sets, *car*, *ecoli*, *hayes-root*, and *tic-tac-toe*, which can be obtained from *MLRepository* [14]. We have used these sets as they are well-known and by using them we can compare the new splitting criteria with classic Entropy. The following table is a brief resume of their characteristics.

Table 1. Characteristics of standard experiences sets. Car.=Cardinal; A.=Attributes; Cl.=Classes; I.E.=Initial Error.

Name	Car.	A.	Types	Subject	Cl.	I.E.
car	1728	6	Symbolic	Automobile	4	29.98
ecoli	336	7	Numerical	Protein	8	57.45
hayes-root	132	5	Symbolic	Character	3	61.37
tic-tac-toe	958	9	Symbolic	Game	2	34.66

Each numerical attribute has been divided into several intervals of similar size according to the range of values.

The experimental results have been obtained by ten cross-validation [4][7]. The average of the success index (SI), the marginal improvement (Marginal), the number of rules (Rules), and the number of nodes (Nodes), for each splitting criterion (S.C.) are shown in the following tables (Tables 2--5).

Table 2. Experimental results for *car* set. S.C.=Splitting Criterion; SI=Success Index.

S.C.	SI	Marginal	Rules	Nodes
Entropy	89.35	64.47	269.7	375.9
b1	85.94	53.10	343.1	494.3
b2	89.35	64.47	269.7	374.8
b3	89.35	64.47	269.5	374.9
b4	89.47	64.87	270.1	374.1

Table 3. Experimental results for *ecoli* set. S.C.=Splitting Criterion; SI=Success Index.

S.C.	SI	Marginal	Rules	Nodes
Entropy	72.60	52.30	51.0	121.8
b1	72.33	51.83	52.8	119.2
b2	73.22	53.38	51.4	121.0
b3	73.51	53.89	51.9	120.7
b4	73.21	53.36	50.4	121.1

Table 4. Experimental results for *hayes-root* set. S.C.=Splitting Criterion; SI=Success Index.

S.C.	SI	Marginal	Rules	Nodes
Entropy	62.97	39.66	35.6	50.8
b1	65.22	43.32	35.8	52.7
b2	63.74	40.91	35.7	51.1
b3	65.28	43.42	35.9	51.3
b4	65.28	43.42	35.9	51.3

Table 5. Experimental results for *tic-tac-toe* set. S.C.=Splitting Criterion; SI=Success Index.

S.C.	SI	Marginal	Rules	Nodes
Entropy	81.31	46.06	181.6	285.0
b1	83.39	52.07	212.4	336.4
b2	82.04	48.18	177.9	279.9
b3	82.04	48.18	177.9	279.9
b4	82.04	48.18	177.9	279.9

b2 and *b3* have an SI better than *Entropy* in three of four sets, and they have an SI as good as *Entropy* in the other set; *b4* has an SI better than *Entropy* in all sets; *b1* has SI values near to *Entropy* in one set, and better than *Entropy* in two other sets; the core advantage is simplicity of computation.

5 Conclusions

A new family of splitting criteria based upon the "*b*est class" concept has been defined. They are permissible, easy to compute (no logarithms appear in their definition), allow us to work with experiences with fuzzy classes, and produce good empirical results. *b1* is very easy to compute. This ease of computation could be of more importance than a better SI in some applications. *b2*, *b3*, and *b4* can be computed more easily than *Entropy*, and the success indexes are better, with a similar number of rules. For these reasons we think that this family of splitting criteria should be considered when deciding the appropriate splitting criterion for a concrete problem.

References

1. Breiman, L., Friedman, J. H., Olshen, R. A., Stone, C. J., Classification and Regression Trees. Belmont, CA: Wadsworth International Group (1984)
2. Buntine, W. and Nibblett, T. A., Further Comparison of Splitting Rules for Decision-Tree Induction. Machine Learning 8 (1992) 75-85
3. Chen, Y. H., Wang, W. J., Fuzzy entropy management via scaling, elevation and saturation. Fuzzy Sets and Systems 95 (1998) 173-178
4. Efron, B., Estimating the error rate of a prediction rule: Improvement on cross-validation. Journal of the American Statistical Association 78 (1983) 316-331
5. Fayyad, Usama, Irani, Keki, Technical Note. On the Handling of Continuous-Valued Attributes in Decision Tree Generation. Machine Learning 8 (1992) 87-102
6. Freund, Yoav, Boosting a Wek Learning Algorithm by Majority. Information and Computation 121 (1995) 256-285
7. Goutte, C., Note on free lunches and cross-validation. Neural Computation 9 (1997) 1211-1215

8. Hong, Tzung-Pei, Lee, Chai-Ying, Induction of fuzzy rules and membership functions from training examples. Fuzzy Sets and Systems 84 (1996) 33-47
9. Hunt, E. B., Marin, J. and Stone, P. T., Experiments in induction. Academic Press (1996)
10. Kearns, Michel, Mansour, Yishay, On the Boosting Ability of Top-Down Decision Tree Learning Algorithms. Journal of Computer and System Sciences 58 (1999) 109-128
11. López de Mántaras R., Technical Note. A Distance-Based Attribute Selection Measure for Decision Tree Induction. Machine Learning 6 (1991) 81-92
12. Mesiar, Radko , Rybárik, Ján, Entropy of fuzzy partitions: A general model. Fuzzy Sets and Systems 99 (1998) 73-79
13. Michalski, R., Unifying principles and a methodology of Inductive learning. Artificial Intelligence (1983)
14. MLRepository http: //www.ics.uci.edu/˜mlearn/MLRepository
15. Quinlan, J.R., Discovering rules from large collections of examples: a case study. Expert Systems in the MicroElectronic Age. Edinburgh University Press (1979)
16. Quinlan, J.R., Induction of Decision Trees. Machine Learning 1 (1986) 81-106
17. Quinlan, J.Ross, C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, Inc. (1993)
18. Ramos Jiménez, G. , Morales Bueno, R., Formalización de los algoritmos TDIDT y CIDIM. Informe Técnico de Investigación LCC-ITI 99/01. Departamento de Lenguajes y Ciencias de la Computación. Universidad de Málaga (1999)
19. Ramos-Jiménez, G. , Morales-Bueno R. , Villalva-Soria, A., CIDIM. Control of Induction by Sample Division Method. Proc. of IC-AI´2000, International Conference on Artificial Intelligence. Las Vegas, Nevada (USA). CSREA Press (2000) 1083-1087
20. Schlimmer, J. C., Ficher, D., A case study of incremental concept induction. Proc. of the Fifth National Conference on Artificial Intelligence. Morgan Kaufmann Publishers, Inc. (1986) 496-501
21. Shannon, C. E., The mathematical theory of communication. The Bell Systems Technical Journal 27 (1948) 379-423, 623-656
22. Utgoff, Paul, Incremental Induction of Decision Trees. Machine Learning 4 (1989) 161-186
23. Yuan, Yufei , Shaw, Michael J., Induction of fuzzy decision trees. Fuzzy Sets and Sy stems 69 (1995)