Journal of Computer Science and Cybernetics, V.32, N.4 (2016), 1–
DOI 10.15625/1813-9663/30/4/8801

ALGORITHM TO BUILD FUZZY DECISION TREE FOR DATA CLASSIFICATION PROBLEM BASED ON FUZZINESS INTERVALS MATCHING

LE VAN TUONG LAN[1], NGUYEN MAU HAN,[1] AND NGUYEN CONG HAO[2]

[1]*Information Technology Faculty,*
Hue University of Sciences, Hue University, VietNam
lvtlan@yahoo.com, nmhan2009@gmail.com
[2]*Department of Testing, Hue University, VietNam; nchao@hueuni.edu.vn*

Crossref Similarity Check
Powered by iThenticate

Abstract. Nowadays, with the demand to reflect the real world, we have a number of imprecise stored business data warehouses. The precise data classification cannot solve all the requirements. Thus, the fuzzy decision tree classification problem is important for the fuzzy data mining problem. The fuzzy decision classification based on the fuzzy set theory has some limitations derived from the innerself its. The hedge algebra with many advantages has become a really useful tool for solving the fuzzy decision tree classification. However, the sample data homogenising process based on the quantitative methods of the hedge algebra still has some restrictions because of errors evolved and the resulting tree is not truly flexible. So, the fuzzy decision tree obtained is not always highly predictable. In this paper, using fuzziness intervals matching with hedge algebra, the authors proposed an inductive learning method "HAC4.5 fuzzy decision tree" to obtain a fuzzy decision tree with high predictability.

Keywords. Hedge algebra, data mining, fuzziness intervals matching, fuzzy decision tree, HAC4.5.

1. INTRODUCTION

The real world is infinite while our language is limited, and there inevitably appear phrases that are inexact or ambiguous. Therefore, in practice, the business data warehouses fuzzily stored are inevitable, so the precise data classification can not solve all the requirements. The fuzzy classification problem has been studied by many scientists with different approaches [1–3, 6, 17, 18, 20–23, 25–27], including fuzzy decision tree classification which is of great interest due to the intuitiveness and effectiveness of the training model.

1. Zadeh, Chang, Fuller, Hesham, Ishibuchi, Lee George, Wang, Lee, Wei-Yuan Cheng, Chia-Feng Juang, etc. [4, 5, 7–9, 11, 17, 24, 29–32] has built the fuzzy decision tree based on the fuzzy set theory. They have provided many solution approaches based on the fuzzy set theory combined with neural networks, genetics, support vector machines to solve the limitations of the precise classification problems. However, the shortcomings derived from the inner nature of the fuzzy set theory still remain.

- It is difficult to simulate the complete language structure that human use for reasoning. The ordered structure induced from the fuzzy concepts with the linguistic value is not indicated on the fuzzy set.

© 2016 Vietnam Academy of Science & Technology

- In the reasoning process, sometimes it is necessary to approximate the linguistic, i.e to find a linguistic value whose meaning is approximated with a given fuzzy set.This causes the complexity and errors in the approximation process that depend on the subjectivity.

2. Zengchang, Jonathan Lawry, Yongchuan Tang, etc. have set the linguistic values for the fuzzy data set and built a linguistic decision tree (LDT) using the approach of the ID3 algorithm of the precise decision tree with the nodes corresponding to the linguistic attributes (LID3) [9, 13, 15, 24]. However,

- This approach will give rise to a multilevel tree with a large horizontal division at the linguistic nodes when the linguistic values set of the fuzzy attribute is large (Figure 1), hence leading to overfitting. In addition, at this node, it is imposible to use the binary division of the C4.5 algorithm because there is no order between the linguistic values.

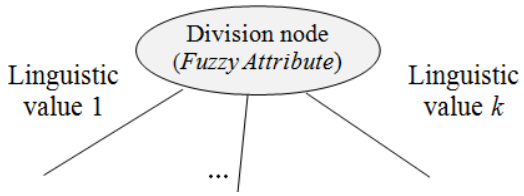

Figure 1. Multilevel position according to linguistic values at fuzzy attribute

Table 1. Mushroom data

Cap Shape	Cap Surface	CapColor	Bruises	Odor	Gill Attachment	Gill Spacing	Gill Size	Gill Color	Stalk Shape	Stalk Root	Stalk Surface Above Ring	Stalk Surface Below Ring	Stalk Color Above Ring	Stalk Color Below Ring	Veil Type	Habitat	Veil Color	Ring Number	Ring Type	Spore Print Color	Population	**Classes**
x	s	n	t	p	f	c	n	k	e	e	s	s	w	w	p	78	w	o	p	k	25	poisonous
x	s	y	t	a	f	c	b	k	e	c	s	s	w	w	p	55	w	o	p	n	25	edible
b	s	w	t	l	f	c	b	n	e	c	s	s	w	w	p	34	w	o	p	n	More Dense	edible
x	y	w	t	p	f	c	n	n	e	e	s	s	w	w	p	78	w	o	p	k	50	poisonous
x	s	g	f	n	f	w	b	k	t	e	s	s	w	w	p	78	w	o	e	n	25	edible
x	y	y	t	a	f	c	b	n	e	c	s	s	w	w	p	55	w	o	p	k	25	edible
b	s	w	t	a	f	c	b	g	e	c	s	s	w	w	p	Possibly Dry	w	o	p	k	15	edible
b	y	w	t	l	f	c	b	n	e	c	s	s	w	w	p	78	w	o	p	n	10	edible
x	y	w	t	p	f	c	n	p	e	e	s	s	w	w	p	Possibly Dry	w	o	p	k	15	poisonous
b	s	y	t	a	f	c	b	g	e	c	s	s	w	w	p	78	w	o	p	k	Rare	edible
x	y	y	t	l	f	c	b	g	e	c	s	s	w	w	p	55	w	o	p	n	15	edible
x	y	y	t	a	f	c	b	n	e	c	s	s	w	w	p	78	w	o	p	k	15	edible
b	s	y	t	a	f	c	b	w	e	c	s	s	w	w	p	34	w	o	p	n	Less Rare	edible
x	y	w	t	p	f	c	n	k	e	e	s	s	w	w	p	78	w	o	p	n	50	poisonous
x	f	n	f	n	f	w	b	n	t	e	s	f	w	w	p	Less Dry	w	o	e	k	1	edible
s	f	g	f	n	f	c	n	k	e	e	s	s	w	w	p	78	w	o	p	n	50	edible
f	f	w	f	n	f	w	b	k	t	e	s	s	w	w	p	Very Wet	w	o	e	n	15	edible
x	s	n	t	p	f	c	n	n	e	e	s	s	w	w	p	More Dry	w	o	p	k	15	poisonous
x	y	w	t	p	f	c	n	n	e	e	s	s	w	w	p	34	w	o	p	n	15	poisonous
x	s	n	t	p	f	c	n	k	e	e	s	s	w	w	p	55	w	o	p	n	15	poisonous
b	s	y	t	a	f	c	b	k	e	c	s	s	w	w	p	78	w	o	p	n	10	edible
x	y	n	t	p	f	c	n	n	e	e	s	s	w	w	p	Very Dry	w	o	p	n	15	poisonous
b	y	y	t	l	f	c	b	k	e	c	s	s	w	w	p	55	w	o	p	n	25	edible
b	y	w	t	a	f	c	b	w	e	c	s	s	w	w	p	Wet	w	o	p	n	15	edible
b	s	w	t	l	f	c	b	g	e	c	s	s	w	w	p	78	w	o	p	k	10	edible
f	s	w	t	p	f	c	n	n	e	e	s	s	w	w	p	55	w	o	p	n	25	poisonous
x	y	y	t	a	f	c	b	n	e	c	s	s	w	w	p	55	w	o	p	n	Very Dense	edible
x	y	w	t	l	f	c	b	w	e	c	s	s	w	w	p	34	w	o	p	n	Less Dense	edible
f	f	n	f	n	f	c	n	k	e	e	s	s	w	w	p	20	w	o	p	k	More Rare	edible

- Furthermore, with the precise values in the fuzzy attribute domain of the training data set, a sub-interval of the precise values will be mapped to become a linguistic value, resulting in more errors.

For example, with Mushroom training data (in Table 1), the classification of Mushroom for the *Habitat* and *Population* attributes has many errors due to the fact that the training data contain both precise and imprecise data.

3. An approach based on hedge algebra proposed since 1990 by Ho and Wechler has

several advantages because, in this approach, each linguistic value of a linguistic variable is an element of the hedge algebraic structure, so it can be matched.

In the hedge algebraic approach, there are homogeneous fields whose data include both precise data and imprecise data. Ho, Hao, Viet, Son, Long, Nam, ect. [10], [12–16, 19, 28] have indicated a semantic point-based quantitative method to homogenise the data in terms of number value or linguistic value and how to query the data on this attribute. Therefore, the classification on the homogeneous sample set can be learned.

The problem of constructing a fuzzy decision tree can use the algorithms for constructing the precise decision tree such as C4.5, SLIQ, ect. to learn [21, 22, 25, 26] with the binary division nodes calculated according to the division points with the linguistic values that are ordered and completely determined with a corresponding number value in the constructed hedge algebra.

However, the homogenization process based on the point-based semantic quantitative method has some errors because a sub-interval of existing precise values will be attributed to a point, i.e. a corresponding linguistic values; this also causes approximate values to appear that can be partitioned in two different sub-intervals, resulting in the difference of data classification. In addition, it is also difficult to predict from the resulting tree in cases in which the prediction is neccessary, and there is an overlap at the fuzzy division point. For example, it is neccessary to predict for the sub-interval $[x_1, x_2]$, in which $x_1 < x$ and $x_2 > x$ at the fuzzy division node in Figure 2.

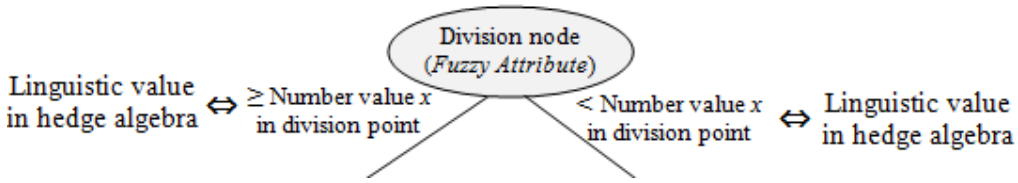

Figure 2. Binary division point in terms of the linguistic value or number value at the fuzzy attribute with the point-based semantic quantitative method of hedge algebra

In this paper, a fuzzy decision tree learning method with a heterogeneous training sample set is proposed. This method based on the fuzziness interval matching method in order to retain the precise value domain while still matching the fuzzy values in the sample training set with the purpose to minimize errors in the prediction process.

The paper is organized as follows: In the second section, the fuzzy interval matching method will be recalled. In section 3, the improvement from the HAC4.5 algorithm for fuzzy data classification will be proposed. Section 4 will be devoted to experiment and discussion. Some conclusions will be given in Section 5.

2. CONSTRUCTING FUZZINESS INTERVALS MATCHING METHOD BASED ON HEDGE ALGEBRA

Hedge algebra is an approach to detect the algebraic structure of the value domain of the linguistic variable. In view of algebra, each value domain of the linguistic variable X can be interpreted as an algebra $\underline{X} = (X, G, H, \leq)$, in which $Dom(X) = X$ is the term domain of linguistic variable X generated from a set of primary generators $G = \{c^-, c^+\}$ by the impact of the hedges $H = H^- \cup H^+$; W is a neutral element; $\leq$ is a semantically ordering relation

on X, it is induced from the natural qualitative meaning of terms. Order structure induced directly so there is the difference compared to other approaches. When adding some special elements, then hedge algebra becomes an abstract algebra $\underline{X} = (X, G, H, \Sigma, \Phi, \leq)$, which Σ, Φ are two operators taking the limit of the set term generated when affected by the hedges in H. Alternatively, if the symbol $H(x) = \{h_1...p x | h_1, ..., h_p \in H\}$, then $\Phi_x = infimum\ H(x)$ and $\Sigma_x =\ supremum\, H(x)$. Thus, hedge algebra X is built on foundation of hedge algebra $\underline{X} = (X, G, H, \leq)$, where $X = H(G)$, Σ and Φ are two additional operators. Then $X = X \cup lim(G)$ with $lim(G)$ is the set of elements limited: $\forall x \in lim(G), \exists u \in X : x = \Phi_u$ or $x = \Sigma_u$. The limitation elements are added to hedge algebra $\underline{X}$ to make the new calculation meant and so $\underline{X} = (X, G, H, \Sigma, \Phi, \leq)$ called complete hedge algebra. The quantitative semantics function (ν), fuzziness measure function (fm), sign function (SGN) and the properties of hedge algebra can be refered to the relevant documents [10, 16].

2.1. Definition of fuzziness intervals

Definition 1 [10]. A fuzziness interval of $x \in X$ denoted by $I(x)$ is a sub-interval of $[0, 1]$ and has a length determined by the fuzziness measure of x, i.e. $fm(x) = |I(x)|$.

For every term x, the fuzziness interval of $x \in X$ is a sub-interval of $[0, 1]$ of length $fm(x)$, denoted by $I_{fm}(x)$, which will be constructed by induction on the length of x as follows:

i) For x of length 1, i.e. $x \in \{c^+, c^-\}$, $I_{fm}(c^+)$ and $I_{fm}(c^+)$ are intervals which constitute a partition of $[0, 1]$ and satisfy the conditions that $c^- \leq c^+$. This implies $I_{fm}(c^-) \leq I_{fm}(c^+)$, $|I_{fm}(c^+)| = fm(c^+)$ and $|I_{fm}(c^+)| = fm(c^+)$, where $|I(x)|$ denotes the length of $I(x)$, and the notation $U \leq V$ means that for $\forall x \in U, \forall y \in V$, it yields $x \leq y$.

ii) Suppose that $I_{fm}(x)$ has been defined and $|I_{fm}(x)| = fm(x)$, for all x of length $k\,(l(x) = k)$. Then, the fuzziness intervals $y = h_i x$, $\forall i \in [-p, -p + 1, ..., -1, 1, 2, ..., q]$ (then l(y) = k + 1) are the set $\{I_{fm}(h_i x)\}$ constructed so that they constitute a partition of $I_{fm}(x)$ and satisfy the conditions that $|I_{fm}(h_i x)| = fm(h_i x)$ and set $\{I_{fm}(h_i x)\}$ is a linearly ordered set, whose order is induced by that of the set $\{h_{-q}x, h_{-q+1}x, ..., h_p x\}$.

When $l(x) = k$, $I(x)$ denoted as $I_{fm}(x)$, $X_k = \{\forall x \in X : l(x) = k\}$ is the set of elements in X that has length equal to k, $I_k = \{I_k(x) : x \in X_k\}$ is the set of fuzziness interval level k.

Definition 2. Two fuzziness intervals are equal, denoted $I(x) = I(y)$, if they are determined by the same value $(x = y)$, i.e. $I_L(x) = I_L(y)$ and $I_R(x) = I_R(y)$, where $I_L(x)$ and $I_R(x)$ are the most left and right point of the fuzziness interval $I(x)$. Otherwise, $I(x) \neq I(y)$ is denoted.

Theorem 1 [10]. *Let $\underline{X} = (X, G, H, \leq)$ be a hedge algebra, we have:*

i) *If $sign(h_p x) = +1$, then*

$$I(h_{-q}x) \leq I(h_{-q+1}x) \leq ... \leq I(h_{-1}x) \leq I(h_1 x) \leq I(h_2 x) \leq ... \leq I(h_p x)$$

and if $sign(h_p x) = -1$, then

$$I(h_{-q}x) \geq I(h_{-q+1}x) \geq ... \geq I(h_{-1}x) \geq I(h_1 x) \geq I(h_2 x) \geq ... \geq I(h_p x)$$

.

ii) *The set $I_k = \{I_k(x) : x \in X_k\}$ is a partition of $[0, 1]$.*

iii) *For each m is a positive integer, the set $\{I(y) : y = h_m h_{m-1}...h_1 x\, \forall h_m h_{m-1}...h_1 \in H\}$ is a partition of then fuzziness interval $I(x)$.*

iv) *The set* $I_k = \{I_k(x) : x \in X_k\}$ *is smoother than the set* $I_{k-1} = \{I_k(x) : x \in X_{k-1}\}$, *i.e. for each interval of* I_k *is a part of interval* I_{k-1}.

v) *If* $x < y$ *and* $l(x) = l(y) = k$, *then* $I_k(x) \leq I_k(y)$ *and* $I_k(x) \neq I_k(y)$.

Proposition 1. $\forall x, y \in X$, *we determine two fuzziness intervals* $I_k(x)$ *and* $I_l(y)$. *And then they are either without an inheriting relation, or related to each other if* $\exists I_v(z) \in I_v, v \leq \min(l,k), I_L(z) \leq I_L(y), I_R(z) \geq I_R(y)$ *and* $I_L(z) \leq I_L(x), I_R(z) \geq I_R(x)$, *i.e.* $I_v(z) \supseteq I_k(x)$ *and* $I_v(z) \supseteq I_l(y)$, *i.e.* x, y *are generated from* z, $x = h_{in}...h_{i1}z, y = k_{jm}...k_{j1}z, \forall h_i, k_j \in H$.

2.2. The fuzziness intervals matching

Let $\underline{X} = (X, G, H, \leq)$ be a hedge algebra and an interval value $[a, b]$. To compare a value $x \in X$ with $[a, b]$:

- Change $[a, b]$ into a sub-interval of $[0, 1]$ because the fuzziness of x is the sub-interval of $[0, 1]$.

- To compare a value $x \in X$ with a sub-interval of $[0, 1]$, we only consider the intersection of two corresponding sub-intervals of $[0, 1]$.

From [10], for each $x \in X, I(x) \subseteq [0, 1]$ and $|I(x)| = fm(x)$, $[I_a, I_b] = [f(a), f(b)] \subseteq [0, 1]$ the same to change $[a, b]$ into sub interval of $[0, 1]$.

i) For each $[I_a, I_b]$ if there is $x \in X$ so that $[I_a, I_b] \subseteq I(x)$, then $[a, b] = {}_{|x|}x$, (see Figure 3).

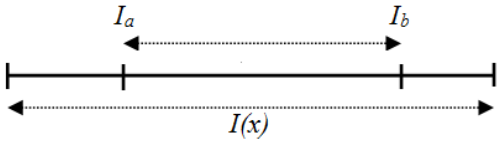

Figure 3. Relationship in case $[I_a, I_b] \subseteq I(x)$

ii) For each $[I_a, I_b]$ so that $[I_a, I_b] \not\subset I(x) \forall x \in X$, and with $x_1 \in X$ and supposed that $x < x_1$, if $|[I_a, I_b] \cap I(x)| \geq |[I_a, I_b]|/\pounds$ then $[a, b] = {}_{|x|}x$, where $\pounds$ is the number of intervals $I(x_i) \subseteq [0, 1]$ so that $[I_a, I_b] \cap I(x_i) \neq \emptyset$, (see Figure 4).

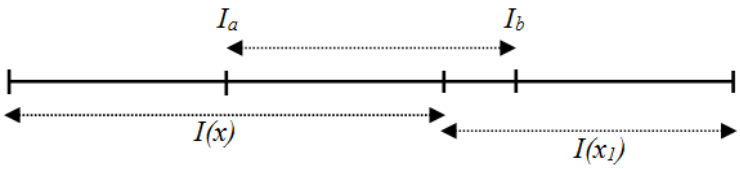

Figure 4. Relationship in case $[I_a, I_b] \subseteq I(x)$.

Otherwise, if $|[I_a, I_b] \cap I(x_1)| \geq |[I_a, I_b]|/\pounds$ then $[a, b] = {}_{|x_1|}x_1$, (see Figure 5).

iii) For each $[I_a, I_b]$ and $x \in X$ so that $[I_a, I_b] \cap I(x) = \emptyset$ then there is $z \in X$ so that $[I_a, I_b] \subseteq I(z)$ and $I(x) \subseteq I(z)$ then $[a, b] = {}_{|z|}x$, (see Figure 6).

Definition 3. Let $[a_1, b_1]$ and $[a_2, b_2]$ be two different precise intervals corresponding to two fuzziness intervals $[I_{a_1}, I_{b_1}]$, $[I_{a_2}, I_{b_2}] \subseteq [0, 1]$. We say that interval $[a_1, b_1]$ preceeds $[a_2, b_2]$ or

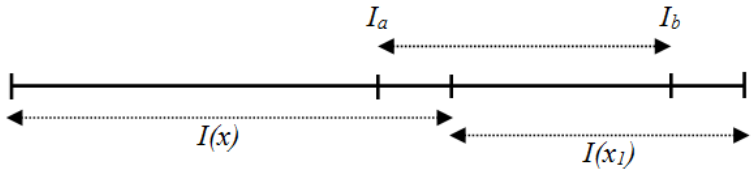

Figure 5. Relationship in case $[I_a, I_b] \not\subset I(x)$

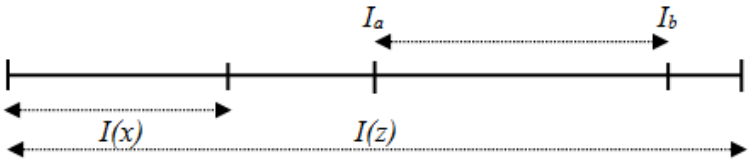

Figure 6. Relationship in case $[I_a, I_b] \cap I(x) = \emptyset$

$[a_2, b_2]$ follows $[a_1, b_1]$, written as $[a_1, b_1] < [a_2, b_2]$ *or* $[I_{a_1}, I_{b_1}] < [I_{a_2}, I_{b_2}]$ if:

i) $b_2 > b_1$ (i.e. $I_{b2} > I_{b1}$);

ii) if $I_{b_2} = I_{b_1}$ (i.e. $b_2 = b_1$) then $I_{a_2} > I_{a_1}$ (i.e. $a_2 > a_1$).

Now, we say that the sequence of intervals $[a_1, b_1]$, $[a_2, b_2]$ is the sequence having pre-order and post-order relations.

Theorem 2. *Let* $[a_1, b_1]$, $[a_2, b_2]$, ..., $[a_k, b_k]$ *be k different paired intervals. Then, it always yields a sequence of k intervals with post-preorder relations.*

Proof

Clearly, for k different paired intervals, such as $[a_1, b_1], [a_2, b_2], ..., [a_k, b_k]$, we always find a first interval $[a_i, b_i]$ of the sequence, where $a_i = \min(a_1, a_2, ..., a_n)$.

If there are many intervals $[a_j, b_j]$, $i = 1..k$ and $a_j = a_i$ then we select $[a_i, b_i]$ as an interval so that b_i is the smallest value of b_j. The selection b_i is always unique because the given intervals are different from each other. Thus, if $a_i = a_j$, then $b_i \neq b_j$ (Definition 2).

After having the very first interval $[a_i, b_i]$ of the sequence, we continue to find the second interval, etc. After k steps of finding and sorting, it yields the sequence with k intervals, and the elements of the sequence are sorted according to the post-preorder relation. ■

3. HAC4.5 ALGORITHM FOR FUZZY DECISION TREE DATA CLASSIFICATION PROBLEM

3.1. Introduction

The C4.5, an algorithm improved by Quinlan [9], calculates infomation gain to look for the division points. The attribute, after being chosen for data classification, is classfied according to its different values if it is discrete; otherwise, it is neccesary to find a threshold to split two sub-sets according this threshold if the attribute is continuous.

Because the fuzzy attribute of the training sample set partitioned according to the fuzzy interval is a sub-interval of $[0, 1]$, and the domain of its values is sorted linearly according to the post-preorder relation, we can compare to find the threshold of this set of valued at any interval $I(x) = [I_a, I_b] \subseteq [0, 1]$ as the continuous number values in the C4.5 algorithm.

Finding the threshold to split is also based on the information gain ratio of thresholds in set D at that node. The information gain ratio of thresholds for attribute A is number attribute in D at that node.

Suppose that attribute A is a fuzzy attribute partitioned according to the fuzzy interval and there are k different intervals already sorted according to the post-preorder relation: $[I_{a_1}, I_{b_1}] < [I_{a_2}, I_{b_2}] < ... < [I_{a_k}, I_{b_k}]$.

It yields that k thresholds are computed: $Th_i^{HA} = [I_{a_i}, I_{b_i}], (1 \leq i < k)$. At each threshold Th_i^{HA}, the data set D of this node is divided into two sub-sets: $D_1 = \{\forall [I_{a_j}, I_{b_j}] \,|[I_{a_j}, I_{b_j}] \leq Th_i^{HA})\}$ *and* $D_2 = \{\forall [I_{a_j}, I_{b_j}] \,|[I_{a_j}, I_{b_j}] > Th_i^{HA})\}$.

Then, we have:

$$\text{Gain}^{HA}(D, Th_i^{HA}) = \text{Entropy}(D) - \frac{|D_1|}{|D|} \times \text{Entropy}(D_1) - \frac{|D_2|}{|D|} \times \text{Entropy}(D_2),$$

$$\text{SplitInfo}^{HA}(D, Th_i^{HA}) = -\frac{|D_1|}{|D|} \times \log_2 \frac{|D_1|}{|D|} - \frac{|D_2|}{|D|} \times \log_2 \frac{|D_2|}{|D|},$$

$$\text{GainRatio}^{HA}(D, Th_i^{HA}) = \frac{\text{Gain}^{\text{HA}}(\text{D}, \text{Th}_\text{i}^{\text{HA}})}{\text{SplitInfo}^{\text{HA}}(\text{D}, \text{Th}_\text{i}^{\text{HA}})}.$$

Based on the computed information gain ratio of thresholds, we select the threshold whose information gain ratio is the biggest to split D into two sub-sets.

3.2. The HAC4.5 algorithm

Input: Training data set D.
Output: Fuzzy decision tree S.
Method:
For each (fuzzy attribute X in D)
  Begin
    Built a hedge algebra $\underline{X_k}$ corresponding with fuzzy attribute X;
    Transform number values and linguistic values of X into intervals $\subseteq [0, 1]$;
  End;
Set of leaf node S; $S = D$;
For each (leaf node L in S)
  If (L homogenise) or (L set of attribute is empty) then
    L.Label = Class name;
  Else
    Begin
      X is attibute having GainRatio or GainRatioHA as the biggest;
      L.Label = Attribute name X;
      If (L is fuzzy attribute) Then
        Begin
          T = Threshold have $GainRatio^{HA}$ as the biggest;
          $S_1 = \{I_{x_i} | I_{x_i} \subseteq L, I_{x_i} \leq T\}$;
          S_1.Father node = L;
          S_1.Attribute = L.Attribute $-X$;
          $S_2 = \{Ix_i | Ix_i \subseteq L, Ix_i > T\}$;
          S_2.Father node = L;

S_2.Attribute = L.Attribute - X;
$S = S + S_1 + S_2 - L$;// Mark the reviewed L button and add two child
End
Else
If (L is continuous attribute) then
Begin
T = Threshold have GainRatio as the biggest;
$S_1 = \{x_i | x_i \in L, x_i \leq T\}$;
S_1.Father node = L;
S_1.Attribute = L.Attribute - X;
$S_2 = \{x_i | x_i \in L, x_i > T\}$;
S_2.Father node = L;
S_2.Attribute = L.Attribute - X;
$S = S+S_1+S_2-L$; // Mark the reviewed L button and add two
End
Else { L is discrete attribute }
Begin
$P = \{x_i | x_i \in K, x_i single\}$;
For (each $x_i \in P$) do
Begin
$S_i = \{x_j | x_j \in L, x_j = x_i\}$;
S_i.Father node = L;
S_i.Attribute = L.Attribute $-X$;
$S = S + S_i$;
End;
$S = S-L$; // Mark the reviewed L button and add all child
End;
End;

3.3. Evaluating algorithm

Let m be the number of attributes, n be the number of the training sample. Then the complexity of the C4.5 algorithm is $O(m \times n \times \log n)$. In the instances of HAC4.5 algorithm, first, the complexity of the algorithm calculating the fuzzy interval partitions is $O(n^2)$, after that at a loop step with attribute m_i, if m_i

The accuracy of the algorithm is inferred from the accuracy of the C4.5 algorithm and the matching method in section 2.

Because of using idea of the C4.5 algorithm at this division node, there are no partitions with partial k and the horizontal spread is avoided leading to "overfitting" on the result tree. The additional cost $O(n)$ in the training process is acceptable. Moreover, the training process is performed only once and used to predict several times. Due to the fact that partitioning in the training process is based on the concept of interval partition correlation, so the fuzzy decision tree to be obtained can be used to predict in the case of points or intervals making the prediction convenient.

4. EXPERIMENTAL EVALUATION

The experimental program is implemented in the Java language (Eclipse Mars Release (4.5.0) anh run on a computer with the following configuration: Processor Intel $Core^{TM}$i5-2450 CPU @2.50GHz (4CPUs), 2.50 GHz, RAM 4GB, System type 64 bit for all the three algorithms: the C4.5, point-based homogenization matching, and interval matching with HAC4.5 on two training sample sets, namely Mushroom and Adult.

- In the Mushroom training sample set there are more than 8000 records containing 22 attributes, in which attributes Habitat and Population contain both precise data and imprecise data. We use 5000 records for training and randomly select 2000 records from the 3000 remaining records for testing.

- The training sample set Adult has 40000 records with 14 attributes consisting of discrete data, continuous data, logic and imprecise data, in which there are two attributes Age and HoursPerWeek containing precise data and imprecise data. We use 20000 records for the training sample set and in the 20000 remaining records, 5000 records are randomly selected for testing.

4.1. Results of Mushroom data classification

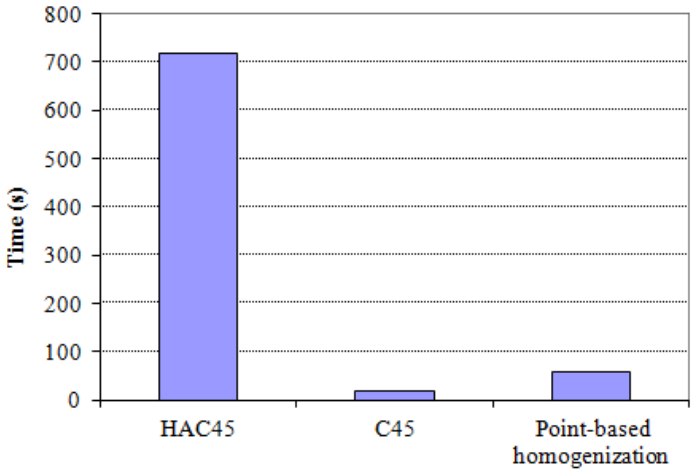

Figure 7. Matching training time in Mushroom sample

Table 2. Training with 5000 Mushroom sample for matching training in Mushroom data

Algorithm	Time (s)
HAC4.5	717.3
C4.5	18.9
Point-based homogenization	58.2

Table 3. Testing ratio from 100 to 2000 Mushroom data sample for matching testing ratio in Mushroom data

Algorithm	100	500	1000	1500	2000
HAC4.5	82.0%	81.0%	86.1%	88.9%	91.5%
C4.5	57.0%	54.8%	51.2%	66.2%	70.0%
Point-based homogenization	71.0%	72.2%	72.6%	77.9%	77.2%

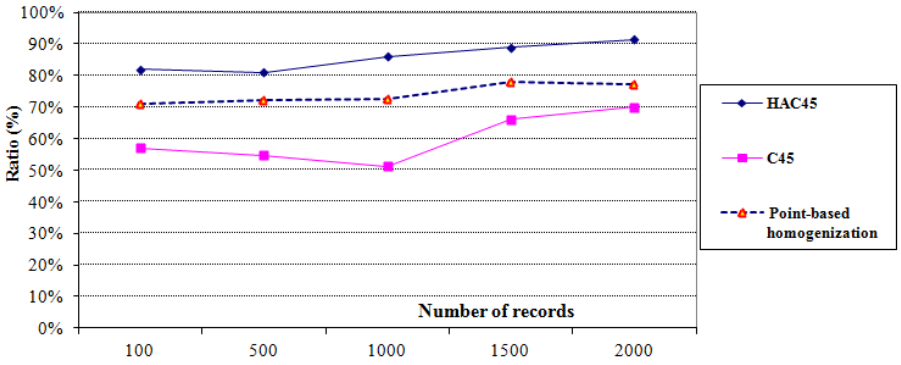

Figure 8. Matching testing ratio from 100 to 2000 in Mushroom data sample

4.2. Results of Adult prediction data

Table 4. Training time in 20000 sample for matching training in Adult data

Algorithm	Time (s)
HAC4.5	1863.7
C4.5	479.8
Point-based homogenization	589.1

Table 5. Matching testing ratio in Adult data

Algorithm	1000	2000	3000	4000	5000
HAC4.5	92.3%	91.5%	93.0%	95.0%	96.1%
C4.5	84.5%	85.7%	85.9%	86.2%	85.7%
Point-based homogenization	87.0%	86.2%	87.4%	87.5%	86.6%

Table 6. Testing time from 1000 to 5000 sample in Adult data for matching testing time in Adult data

Algorithm	1000	2000	3000	4000	5000
HAC4.5	2.4	4.7	7.2	9.7	12.1
C4.5	1.4	2.8	4.1	5.5	6.0
Point-based homogenization	2.2	4.6	7.1	9.2	11.8

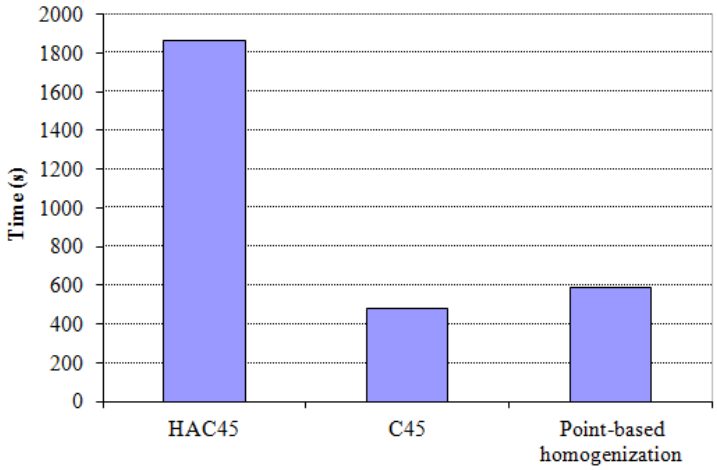

Figure 9. Matching traning time in Adult

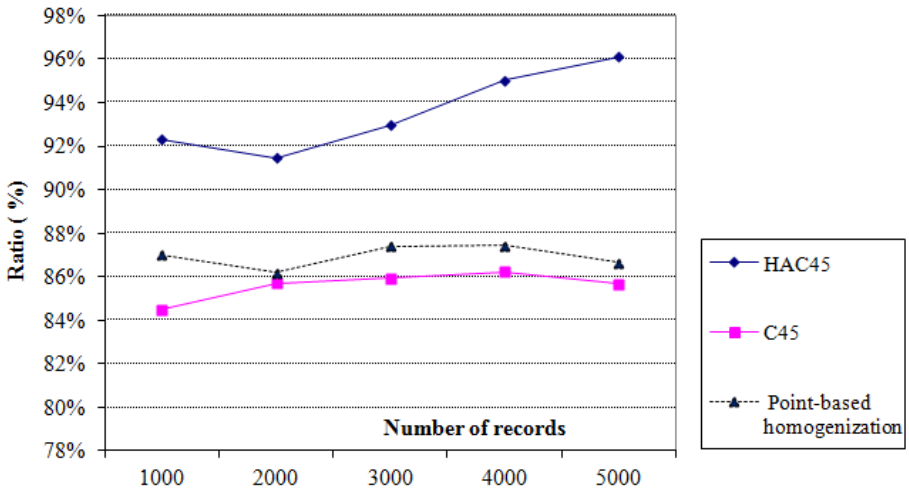

Figure 10. Matching testing ratio from 1000 to 5000 in Adult data sample

4.3. Result evaluation

The results on the data sets Mushroom and Adult derived from using three implemented algorithms C4.5, HAC4.5 and homogenization point matching are as follows:

- **Time:** the C4.5 algorithm is always the fastest for the two samples in terms of training and testing because it ignores the fuzziness values in the sample sets, and process time for these data is not neccessary.

The homogenization of the data set based on the point matching and using this set for the tree training require the construction of the hedge algebras for the imprecise data and the homogenization cost. This algorithm needs more time than the algorithm C4.5.

Because there is a need for the construction of the hedge algebras for the fuzziness fields and the cost for the conversion of the values to the initial sub-interval [0, 1], and at each loop

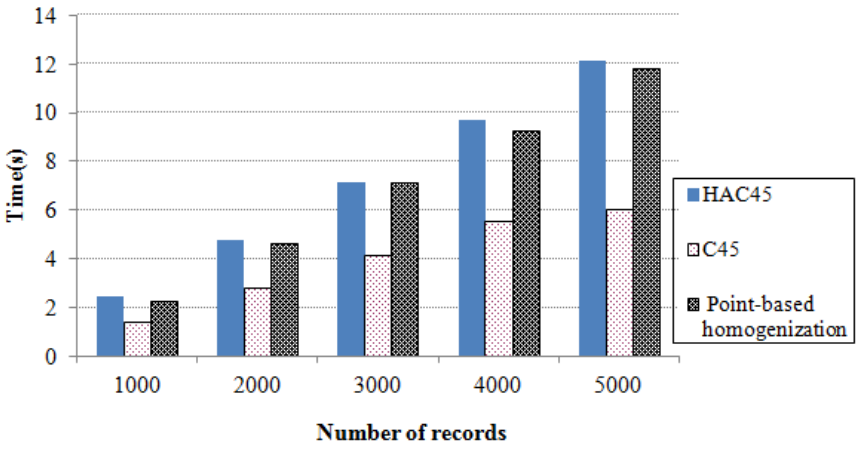

Figure 11. Matching testing time from 1000 to 5000 in Adult data sample

additional time is neccessary for the selection of intervals, the algorithm HAC4.5 is relatively slow compared with other algorithms.

- **The prediction result:** Because the C4.5 algorithm ignores the imprecise values in the sample set, it losses data in the fuzzy attribute, resulting in poor prediction results.

The construction of a hedge algebra for fuzzy attributes and using it to homogenise the training sample set by point matching gives a homogenization training sample set containing precise data and imprecise data. Therefore, the result of the training tree is better, and this algorithm has higher prediction results than C4.5. However, the prediction results in this case are not desirable because the partition of the fuzzy points causes errors in the precise values at the split points.

The prediction results of HAC4.5 is the best because in the tree training, the imprecise values are processed while the precise values remain unchanged, leading to the absence of errors in the partition process.

Although HAC4.5 needs more time for the training, it is an effective method as the result tree has high predictability. Furthermore, the training process is performed only once while the prediction on the result tree is done several times, and thus the processing time of HAC4.5 is acceptable.

5. CONCLUSIONS

The fuzzy decision tree classification problem plays an important role in the process of data mining. However, the fuzzy decision tree classification based on the fuzzy set theory has many disadvantages. The hedge algebra with its numerous advantages has become a really useful tool for solving the decision tree classification problems. Recognizing the limitations of the quantitative semantics methods in the training process, the authors use hedge algebra to propose a fuzzy interval matching method, and on this basis they propose an inductive learning fuzzy decision tree using the algorithm HAC4.5. This algorithm is effective for the decision tree classification problems. The time optimization of the HAC4.5 algorithm will be considered in the future paper.

REFERENCES

[1] J. Abonyi, J. A. Roubos, and M. Setnes, "Learning fuzzy classication rules from labeled data," *Information Sciences*, vol. 150, 2003.

[2] A. Bikas, E. M. Voumvoulakis, and N. D. Hatziargyriou, "Neuro-fuzzy decision trees for dynamic security control of power systems, department of electrical and computer engineering," in *NTUA, Athens, Greece*, 2008.

[3] B. Chandra, "Fuzzy sliq decision tree algorithm," in *IEEE*, 2008.

[4] Chang, R. L. P. Pavlidis, and Theodosios, "Fuzzy decision tree algorithms," in *Man and Cybernetics, IEEE*, 2007.

[5] W. Y. Cheng and C. F. Juang., "A fuzzy model with online incremental svm and margin-selective gradient descent learning for classication problems," in *IEEE Transactions on Fuzzy systems*, 2014, pp. 324–337.

[6] P. Fatima, Parveen, and D. M. Sathik, "Fuzzy decision tree based effective imine indexing," *IJCTEE*, vol. 1, 2011.

[7] R. Fuller, "Neural fuzzy systems," in *Physica-Verlag, Germany*, 1995.

[8] L. C. George and C. T. Lin, "Neural fuzzy systems, a neuro-fuzzy synergism to intelligent systems," *Prentice-Hall International*, 1995.

[9] J. Hang and Honavar, "Learning decision tree classifiers from attribute-value taxonomies and partially specified data," in *Proceedings of the International Conference on Machine Learning. Washington DC*, 2003.

[10] N. C. Hao, "Fuzzy databases with data manipulation based on hedge algebra," Ph.D. dissertation, IOIT, 2008.

[11] H. A. Hefny, A. S. Ghiduk, and A. A. Wahab, "Effective method for extracting rules from fuzzy decision trees based on ambiguity and classifiability," *Universal Journal of Computer Science and Engineering Technology, Cairo University, Egypt.*, vol. 150, pp. 55–63, 2010.

[12] N. C. Ho and N. V. Long, "Fuzziness measure on complete hedges algebras and quantifying semantics of terms in linear hedge algebras," *Fuzzy Sets and Systems*, vol. 158, pp. 452–471, 2007.

[13] N. C. Ho and H. V. Nam, "An algebraic approach to linguistic hedges in zadeh's fuzzy logic," *Fuzzy Sets and Systems*, vol. 129, pp. 229–254, 2002.

[14] N. C. Ho and W. Wechler, "Extended algebra and their application to fuzzy logic," *Fuzzy Sets and Systems*, vol. 52, pp. 259–281, 1992.

[15] N. Ho and W. Wechler, "Hedge algebras: an algebraic approach to structures of sets of linguistic domains of linguistic truth variables," *Fuzzy Sets and Systems*, vol. 35, pp. 281–293, 1990.

[16] N. C. Ho and T. T. Son, "On distance between linguistic values in hedge algebra," *Journal of Computer Science and Cybernetics*, vol. 11, no. 1, pp. 10–20, 1995.

[17] H. Ishibuchi and T. Nakashima, "Effect of rule weights in fuzzy rule-based classification systems," *IEEE Trans. on Fuzzy Systems*, vol. 9, 2001.

[18] F. James, H. Smith, and N. T. Vu, "Genetic program based data mining of fuzzy decision trees and methods of improving convergence and reducing bloat," *Data Mining, Intrusion Detection, Information Assurance*, 2007.

[19] L. V. T. Lan, N. M. Han, and N. C. Hao, "A novel method to build a fuzzy decision tree based on hedge algebras," *International Journal of Research in Engineering and Science*, vol. 4, pp. 16–24, 2016.

[20] D. T. Long, "Method to built fuzzy rule system based on hedge algebra semantic and applied for classification problem," Ph.D. dissertation, IOIT, 2010.

[21] M. Mehta and J. Rissanen, "Sprint: A fast scalable classifier for data mining," in *IBM Almaden Reseach Center*, 1996.

[22] M. Mehta, J. Rissanen, and R. Agrawal, "Sliq: A fast scalable classifier for data mining," in *IBM Almaden Reseach Center*, 1996.

[23] Moustakidis, G. K. S. Mallinis, N. Theocharis, and J. Petridis, "Svm-based fuzzy decision trees for classification of high spatial resolution remote sensing images," in *IEEE*, 2012.

[24] Z. Qin and Y. Tang, "Linguistic decision trees for classification," in *Uncertainty Modeling for Data Mining, Springer*, 2014, pp. 77–119.

[25] J. R. Quinlan, "Simplifying decision trees," *International Journal of Man-Machine Studies*, no. 27, pp. 221–234, 1987.

[26] S. Ruggieri, "Efficient c4.5," in *University Di Pisa*, 2000.

[27] R. H. Tajiri, E. Z. Marques, and B. B. Z., "A new approach for fuzzy classification in relational databases," *Database and Expert Systems Applications, Springer*, pp. 511–518, 2011.

[28] L. X. Viet, "Semanticquantitativelinguistic values of linguistic variable inhedge algebra and applied," Ph.D. dissertation, IOIT, 2009.

[29] T. Wang and H. Lee., "Constructing a fuzzy decision tree by integrating fuzzy sets and entropy," in *ACOS'06 Proceedings of the 5th WSEAS international conference on Applied computer science, World Scientific and Engineering Academy and Society, USA*, 2006, pp. 306–311.

[30] L. A. Zadeh, "Fuzzy sets," *Information and Control 8*, pp. 338–358, 1965.

[31] ——, "Fuzzy sets and fuzzy information granulation theory," in *Beijing Normal University Press, China*, 2000.

[32] Q. Zengchang and J. Lawry, "Linguistic decision tree induction," in *Department of Engineering Mathematics, University of Bristol, United Kingdom*, 2007.

Received on October 21 - 2016
Revised on February 20 - 2017