HUE UNIVERSITY
HUE UNIVERSITY OF SCIENCES

LE VAN TUONG LAN

DATA CLASSIFICATION BY FUZZY DECISION TREE BASE ON HEDGE ALGEBRA

MAJOR: COMPUTER SCIENCE
CODE: 62.48.01.01

SUMMARY OF PHD DISSERTATION

Supervisors:
1. Assoc. Prof. Dr. Nguyen Mau Han
2. Dr.Nguyen Cong Hao

HUE, 2018

INTRODUCTION

1. Rationale of the study

In fact, the fuzzy concept always exists, so the conception of objetcs, which must be used clearly in the classical logic, will not enough to describe the problems of the real world. In 1965, L. A. Zadeh proposed the mathematical formalization of fuzzy concept, since then fuzzy set theory is formed and increasingly attracted the research of many authors. In 1990, NC Ho & W. Wechsler intitated the algebraic aproach to the natural structure of the variable linguisticvaluedomain method. According to this method, each linguistic value of linguistic variable belongs to algebraic topology called hedge algebras. On that basis, there were a lot of authors' studies in many fields of researching: fuzzy control and fuzzy reasoning, fuzzy database, fuzzy classification,...etc.... and had given out many extremely positive results, which is likely to be appilied.

Currently, data mining is a priority problem solved necessarily that data classification is an important process of data mining. It is the process of dividing the data objects into classes based on the characteristics of the data set. The methods commonly used in the learning process classified such as: statistical, neural networks, decision trees ...etc... in which the decision tree is an effective solution. There were a lot of studies to build it but the inductive learning algorithm is the most remarkable such as CART, ID3, C4.5, SLIQ, SPRINT, LDT, LID3,... However, currently, the ways of approaching the data classification learning by a decision tree still have many problems:

- To build a decision tree based on *Entropi* concept of information by traditional methods such as ID3, C4,5, CART, SLIQ, SPRINT,... for the algorithm has a low complex but not high predictability, which may lead to the overfitting problem on the result tree. In addition, these methods can not be used for training and predicting on the sample set containing the value dim, but now the data storage is the inevitable blur on the business data warehouse.

- One approaching is through fuzzy set theory to calculate the informative benefits of the fuzzy attribute for the classification process. This method has solved the imprecise values in the training set through the identification of the dependent function, from which the values can be involved in the training process. Thus, it solved the restriction and

ignored fuzzy data value of classification. However, there are still encountering limitations from intrinsic of fuzzy set theory: the function of themsevles cannot compared to each other, appearing the significant error in the process of approximation, depending on the objective, lacking an linguistic value on the basis of algebra.

- According to the approaching of building decision linguistic tree. Many authors have developed the method of determining the value of the language on the fuzzy data set and built the tree bassed on LID3 method. The construction of the linguistic label for imprecise values based on the probability of the link label while retaining the clear values, this approaching reduces the considerable margin of error for the training process. However, this approaching will generate multicellular tree as there has the large horizontally split in the language button.

- Quantitative methods based on hedge algebra, to homogeneous data on the value or the value of language. The problem of building a decision tree can use the mathematical algorithm according to the decision tree. However, this approaching still has some problems such as: still appear large error when homogeneous according to fuzzy point, difficult in making predictions when there is an overlap in fuzzy devided point of result tree, depending on domain in [ψ_{min}, ψ_{max}] value from the the domain of clearly value of fuzzy.

All algorithms of classification by a decision tree depend mostly on the selection of the training sample set. In the business data warehouse, much of the information services for the prediction, but a large amount of information just means simply storage, servicing the interpreting the information. We make complex model, so increaseing costs for the training process, the more important is that they interfere with the tree and it's the reason why the tree was built without high efficiency. From finding and researching the characteristics and challenges of the problems of the data classification by decision trees, topic: "*Data classification by a fuzzy decision tree based on hedge algebras*" is a major problem to solve.

2. Scope of the study

The thesis focuses on the researching a_model for the learning process from the training set, researching the linguistic value processing methods and building some classification algorithms by the fuzzy decision tree, that resulted highly in prediction and simple to the

users.

3. Research Methodology

The thesis uses synthetic methods, systematization and scientific empirical method.

4. Objectives and content of the thesis

After studying and analyzing the problems of data classification by decision trees of the research in domestically and internationally, the thesis made research objectives as follow:

- Proposing a model to classify by fuzzy decision trees and a method to select the feature training samples set for classification process. Recommending the linguistic value treatment method of inhomogeneous attributes based on hedge algebra.

- Proposing the algorithms by fuzzy decision tree in order to be effective in predicting and simple for users.

To meet the research above objectives, the thesis focused on the following main issues

- Researching some tree algorithms ID3, CART, C4.5, C5.0, SLIQ, SPRINT on each set of training samples to find a suitable learning method.

- Researching the study modeling of the data classification decision tree, building the characteristic selecting method to select the training set for learning decision tree from the business data set.

- Researching to propose the treatment of the linguistic attributes value which is not homogeneous on the sample set based on hedge algebras.

- Recommended some classification algorithms by a fuzzy decision tree that are effective in predicting and simple to users.

5. Scientific and Practical significance

Scientific significance

The main contributions of the thesis about science:

- Building a model of learning data classification by the fuzzy decision trees from training sample set. Recommended a method to select the feature training samples set for classification learning by a decision tree from the data warehouses in order to limit the dependence of experts' opinions in the selection process of training sample set.

- Recommended the treatment process of the linguistic values of inhomogeneous attributes on the training sample set based on the hedge algebras.

- The thesis has built the objective_function of the classification problem by the decision tree, using the order of the linguistic values in hedge algebras. Giving the fuzziness interval matching concepts, the maximum fuzziness interval from that proposed the fuzzy decision tree learning algorithms MixC4.5, FMixC4.5, HAC4.5 and HAC4.5* for classification problem, in order to improve, enhance the accuracy of the data classification learning process by the decision tree for data classification problem.

Practical significance

- To demonstrate the variety application ability of hedge algebras in performing and processing the fuzzy data, uncertain data.

- The thesis contributed to the quantitative problem solving for the linguistic value that does not depend on the domain fixed *Min-Max* value of the classic values of the fuzzy attribute in the sample set.

Based on the concepts of fuzzy intervals and the maximum fuzzy interval, the thesis proposed algorithms for the tree learning process to increase predictability for the data classification problem by decision trees. It makes the learning method for classification problem more variety in generally and classifition by a decision tree in particularly.

- The thesis can use as a reference for Information Technology students, Master students who are researching on classification learning by a decision tree.

6. Structure of the thesis

Apart from the introduction, conclusions and references, the thesis is divided into 3 chapters. *Chapter 1: The theoretical basis of hedge algebras and overview of data classification by the decision trees.* Focusing on analyzing and estimating the recently published research works, point out the existing problems in order to identify the goal and contents needed solving. *Chapter 2: Data classification by a fuzzy decision tree using fuzziness intervals maching points method based on hedge algebras.* Focusing on analyzing the influence of training sample set on the effect of the decision tree. Presenting the methods to select the typical sample for the training process. Analyzing, giving the concept of inhomogeneous sample set, the outlier and constructing the algorithm that can homogenise these attributions.Proposing the algorithms MixC4.5 and FMixC4.5 that are served the decision tree learning process based on inhomogeneous sample set. *Chapter 3: fuzzy decision tree training methods for data*

classification problem based on fuzziness intervals matching. This chapter focussed on researching the decision tree learning process in order to get two followings goals: $f_h(S) \rightarrow max$ and $f_n(S) \rightarrow min$. On the basic of researching the correlation of the fuzziness intervals, this thesis proposes a matching process based on fuzziness intervals and constructs the classification decision tree algorithm based on fuzziness interval HAC4.5 build a quantitative method for the inhomogeneous values, unknown *Min-Max*, of sample set. This thesis also proposes a concept of maximum fuzziness intervals, designs the algorithm HAC4.5* in order to achieve the goal.

The main results of the thesis were reported at scientific conferences and senimar, published in 7 scientific works published in the conferences at home and abroad: one paper is posted in Science and technology magazine at Hue University of Science; another one is posted in the journal Science at Hue University; one paper is posted in Proceedings of the National Workshop FAIR; two papers are posted in the Research, Development and Application of Information Technology & Communications Magazine; one paper is posted in Informatics and Cybernetics journals, one is posted in international IJRES journals.

Chapter 1.
THE THEORETICAL BASIS OF HEDGE ALGEBRAS AND OVERVIEW OF DATA CLASSIFICATION BY THE DECISION TREE

1.1. Fuzzy set theory
1.2. Hedge algebras
1.2.1. The definition of hedge algebras
1.2.2. The measurement function of hedge algebras
1.2.3. Some properties of measurable functions
1.2.4. Fuzziness intervals and the relationship of fuzziness intervals

Definition 1.18. Two the fuzziness intervals are called equal, denoted $I(x) = I(y)$, if they are determined by the same value ($x = y$), i.e. we have $I_L(x) = I_L(y)$ and $I_R(x) = I_R(y)$. Where $I_L(x)$ and $I_R(x)$ are point the tip of the left and right of fuzziness interval $I(x)$. Otherwise, we denoted by $I(x) \neq I(y)$.

Definition 1.19. Let $\underline{X} = (X, G, H, \preceq)$ be a hedge algebra and $x, y \in X$:

1. If $I_L(x) \leq I_L(y)$ and $I_R(x) \geq I_L(y)$ we say that x and y have a correlation $I(y) \subseteq I(x)$, in contrast, we say $I(y) \not\subset I(x)$.

2. When $I(y) \not\subset I(x)$, with $x_1 \in X$ and supposed $x < x_1$, if $|I(y) \cap I(x)| \geq |I(y)|/£$ with £ is the number of inteval $I(x_i) \subseteq [0, 1]$ so that $I(y) \cap I(x_i) \neq \emptyset$, we say that y has a correlation matched to x. Otherwise, if $|I(y) \cap I(x_1)| \geq |I(y)|/£$, we say that y has a correlation matched to x_1.

1.3. Data classification by the decision tree

1.3.1. Classification problem in data mining

U = $\{A_1, A_2, \ldots, A_m\}$ is a set with m attributes, $Y = \{y_1, ..., y_n\}$ is a set of class labels; with $D = A_1 \times ... \times A_m$ is the domain of the respective properties of m, there are n number of layers and N is the number of data samples. Each data $d_i \in D$ belong to $y_i \in Y$ respectively forming pairs $(d_i , y_i) \in (D, Y)$.

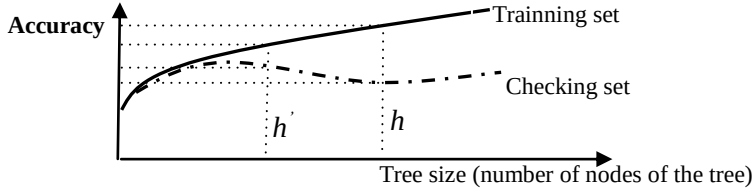
1.3.2. The decision tree

A decision tree is a logical model which represented as a tree, it said the value of a target variable and can be predicted by using the values of a set of predictor variables. We need to build a decision tree, symbol S, to subclass. S acts as a mapping from the data set on the label set, $S : D \rightarrow Y$ (1.4)

1.3.3. Gain information and gain information ratio

1.3.4. The overfiting of the decision tree model

Definition 1.20. A hypothesis h with the model of a decision tree, we say that it is overfitting the set of training data, if there exists a hypothesis h' with h has smaller error it means the accuracy is greater than h' 'on the training data set, but h' has smaller error h on the test data set.



Definition 1.21. A decision tree is called a width spread tree if it exists nodes which have more branches than the multiply of $|Y|$ and its height.

1.4. Data classification by the fuzzy decision tree

1.4.1. The limitations of classification data by the clear decision tree

The goal of this approach is based on training set with the data domains which are identified specifically, building a decision tree with the division obviously follow the value threshold at the division nodes.

♦ **The approach is based on the calculation of gain information attribute:** based on the concept of *Entropy* information to calculate the

gain information and the gain information ratio of the properties at the division time of the training sample set, then select the corresponding attribute that has the maximum information value, as adivision node. If the selected attributes are discrete types, we classify them as distinct values, and if the selected attributes are continuous types, we find the threshold of division to divide them into two subaggregates based on that threshold. Finding the threshold of division based on the thresholds of gain information ratio in training set at that node.

Although this approach gives us the algorithms with low complexity, the division *k-distributed* on the discrete attributes makes the nodes of the tree at a level rose rapidly, increases the width of the tree, leads the tree spread horizontally so it is easy to have an overfittting tree, but difficult to predict.

♦ **The approach is based on the calculation of the coefficient Gini attribute:** based on the calculation of coefficient Gini attributes and coefficient Gini ratio to select a division point for the training set at each moment. According to this approach, we do not need to evaluate each attribute but to find the best division point for each attribute.

However, at the time of dividing the discrete attribute, or always select the division by binary set of SLIQ or binary value of SPRINT so the result tree is unbalanced because it develops the depth rapidly. In addition, each time we have to calculate a large number of the coefficient Gini for the discrete values so the cost of calculation complexity is very high.

In addition, according to the requirements of learning classification by decision tree approach training sample set to be homogeneous and only contains classic data. However, there is always the exitence of fuzzy concepts in the real world so this condition is uncertain of data warehouse. Therefore, the data classsification problem studying by the fuzzy decision tree is a inevitable problem.

1.4.2. Data classification problem by the fuzzy decision tree

Let a classification problem by the decision tree $S: D \rightarrow Y$, in (1.4), if $\exists A_j \in D$ is a fuzzy attribute in D, then (1.4) is a classification problem by the fuzzy decision tree. Decision tree model S have to get high classification result, it means data classification error is the least and the tree has less node but high predictable and there not exits overfitting.

1.4.3. Some problems of data classification problem by the fuzzy decision tree

If we call $f_h(S)$ a effectiveness evaluation function of a predictive process, $f_h(S)$ as a simplicity evaluation function of the tree, the goal of classification problem by the fuzzy decision tree $S : D \rightarrow Y$ is to achieve $f_h(S) \rightarrow max$ and $f_h(S) \rightarrow min$ (1.13).

Two above goals cannot be achieved simultaneously. When the number of tree nodes reduces, it means that the knowledge of the decision tree also reduces the risk of wrong classification increased, but when there are too many nodes that can also cause the information overfitting in the process of classification.

The approaches aim to build the effectiveness decision tree model based on the training set still have some difficulties such as: the ability to predict not high, depending on the knowledge of experts and the selected training samples set, the consistency of the sample set,... To solve this problem, the thesis focused on researching models and decision tree learning solutions based on hedge algebras to training the decision trees effectively.

Chapter 2.
DATA CLASSIFICATION BY A FUZZY DECISION TREE USING FUZZZINESS POINTS MATCHING METHOD BASED ON HEDGE ALGEBRAS

2.1. Introduction

With the goal of $f_h(S) \rightarrow max$ and $f_n(S) \rightarrow min$ of the classificasion problem by the fuzzy decision tree $S : D \rightarrow Y$, we encounter many problems to solve, such as:

1. In business data warehouse, data is stored very multitypes because they serve many different works. Many attributes provide information that is predictable but some attributes cannot be able to reflect the information needed to predict.

2. All inductive learning methods of decision trees such as CART, ID3, C4.5, SLIQ, SPRINT, ... need to the consistency of the sample set. However in the classification problem by the fuzzy decision tree, there is the appearance of the attributes that contains linguistic value, i.e. $\exists A_i \in D$, has a value domain $Dom(A_i) = D_{A_i} \cup LD_{A_i}$, with D_{A_i} is the set of classic values of A_i and LD_{A_i}, the set of linguistic values of A_i. In this

case, the inductive learning algorithm will not process the data sets "*error*" from value domain LD_{A_i}

3. Using the hedge algebras to quantify the linguistic value is often based on the clear value domain of the current attributes, i.e. we can find the value domain[ψ_{min}, ψ_{max}] from the current clear value domain, but it is not always convenient.

2.2. Selecting the characteristic training sample set for classification problem by the decision tree

2.2.1. The characteristic of the attributes in training sample set

Definition 2.1. Attribute $A_i \in D$ called an individual value attribute (separate attribute) if it is a discrete attribute and $|A_i| > (m - 1) \times |Y|$. This set of attributes in D denoted D^*.

Proposition 2.1. The process of constructing a tree if any node based on a discrete attribute then the acquired result may be a spreading tree.

Definition 2.2. Attribute $A_i = \{a_{i_1}, a_{i_2}, \ldots, a_{i_n}\} \in D$ that is between elements a_{i_j}, a_{i_k} with $j \neq k$ does not exist any comparison then we call A_i as a memo attribubute in the sample set, denoted D^G.

Proposition 2.2. If $A_i \in D$ is the memo attribute, we sort out A_i from D without changing the result tree.

Proposition 2.3. If the training sample set contains attribute A_i which is the key of D set, the acquired decision tree will have an overfitting tree at A_i node.

2.2.2 The impact of function dependency between the attributes in the training set

Proposition 2.4. We have a D is sample set with the decision attribute Y, if there is a function dependency $A_i \rightarrow A_j$ and if selected A_i as a division node, its subnodes will not choose A_i as a division node.

Proposition 2.5. We have a D is sample set with the decision attribute Y, if there is a function dependency $A_i \rightarrow A_j$, the received information on A_i is not less than the received information on A_j.

Consequence 2.1. If there is a function dependency $A_1 \rightarrow A_2$ and A_1 is not the key attribute of D then attribute A_2 is not selected as the tree division node.

Algorithmic finding typical training set from business data set

Input: *The sample training set D is selected from business data set;*
Output: *The typical sample training set D*
Algorithm description:

For i = 1 to m do
Begin *Check properties* A_i *;* If $A_i \in \{key, memo\}$ then $D = D - A_i$; End;
i = 1;
While $i < m$ do
Begin $j = i + 1$;
While $j \leq m$ do
Begin If $A_i \rightarrow A_j$ and (A_i *not a key attribute of D)* then $D = D - A_j$
Else If $A_j \rightarrow A_i$ and (A_j *not a key attribute of D)* then $D = D - A_i$;
$j = j + 1$;
End; $i = i + 1$;
End;

2.3. Classification learning by the decision tree based on determining the value attribute domain threshold

2.3.1. The basis of determining the threshold for the learning process

All algorithms are fixed in dividing all discrete attributes of the training set according to binary or *k-distributed*, which makes the result treeinflexible and inefficient. Thus, the need to build a learning algorithm for dividing in a mixture way based on binary *distribution*, *k-distributed* by the attributes to get the tree with reasonable width and depth of the training process.

2.3.2. MixC4.5 algorithm based on the threshold of value domain attribute

Algorithm MixC4.5

Input: *Form D has n sets, m prediction attributes and decisive attributes Y.*
Output: *S decision tree*
Algorithm description:
Choosing particular model (D); The threshold k for attributes;
Create some leaf nodes S; S = D;
For each *(leaf node L belong to S) do*
If (*L homogeneous) or (L is empty*) then *Assign a label for the node with L;*
Else Begin
X = Corresponding attribute GainRatio biggest ; L.label = name of attribute X;
If (*L is constant attribute*) then
Begin *Choosing T proportion to Gain on X;*
$S_1 = \{x_i | x_i \in Dom(L), x_i \leq T\}$; $S_2 = \{x_i | x_i \in Dom(L), x_i > T\}$;
Creating two little buttons for current button which correspond with S_1 *and* S_2 *;*
Marking L button;
End Else *// L is incoherent attribute, divided k-attribute follow C4.5 when* $|L| < k$.
If $|L| < k$ then Begin $P = \{x_i | x_i \in K, x_i$ *unique*$\}$;
For each ($x_i \in P$) do
Begin $S_i = \{x_j | x_j \in Dom(L), x_j = x_i\}$;
Creating a little button i for current button and correspond with S_i;

End; End;
Else Begin //*divided binary follow SPRINT when |L| is over k*
Setting the counting matrix for the values in L;
T = the value in L which have the biggest gain ;
$S_1 = \{x_i | x_i \in L, x_i = T\}$; $S_2 = \{x_i | x_i \in L, x_i \neq T\}$;
Creating two little buttons for current button which correspond with S_1 *and* S_2;
End;
Marking L button;
End; End;

With m is the number of attributes, n is the number of training set, the complexity of the algorithm is $O(m \times n^2 \times log\ n)$. The accuracy and finite of algorithm is derived from algorithms C4.5 and SPRINT.

2.3.3. The experimental implementation and evaluation of algorithms MixC4.5

Table 2.4. Compare the results of training with 1500 samples of MixC4.5 on the Northwind database

Algorithm	Time	Numbers of nodes	Accuracy
C4.5	20.4	552	0.764
SLIQ	523.3	162	0.824
SPRINT	184.0	171	0.832
MixC4.5	**186.6**	**172**	**0.866**

♦ **Training time:** C4.5 always perform *k-distributed* in discrete attributes and remove it at each division step, so C4.5 always achieve the fastest processing speed. The processing time of SLIQ is maximum because of carrying out Gini calculations on each discrete value. Division of MixC4.5 is the mixture between C4.5 and SPRINT, then C4.5 is faster than SPRINT so the training time of MixC4.5 is fairly consistent well with SPRINT.

Table 2.6. Compare the result with 5000 training samples of MixC4.5 on data with fuzzy attribute Mushroom

Algorithm	Training time	The accuracy on the 500 samples	The accuracy on the 1000 samples
C4.5	18.9	0.548	0.512
SLIQ	152.3	0.518	0.522
SPRINT	60.1	0.542	0.546
MixC4.5	**50.2**	**0.548**	**0.546**

♦ **The size of the result tree:** SLIQ carried out the binary dividing based on the set so its nodes are always minimum and C4.5 always divided by *k-distributed* so its nodes are always maximum. MixC4.5

does not homogenise well with SPRINT because the SPRINT algogithm's nodes are less than the C4.5 algogithm's nodes.

♦ **The Prediction Efficiency:** The MixC4.5 improvement is from the combination between C4.5 and SPRINT so the result tree has the predictability better than the other algorithms.However, the match between the training set without fuzzy attribute Northwind and the training set contains fuzzy attribute Mushroom, the predictability of MixC4.5 got a big variance that it could not handle, so it ignored the fuzzyvalues.

2.4. Learning classificationby the fuzzy decision tree based on fuzzy point matching

2.4.1. Construction data classification model by using the fuzzy decision tree

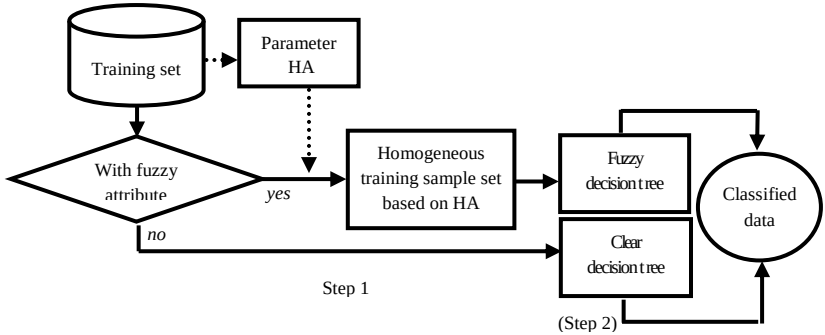


Figure 2.7. A proposal model for classification learning by the fuzzy decision tree

2.4.2. The problem of the inhomogenization training sample set

Definitions 2.4. Fuzzy attribute $A_i \in D$ called an inhomogeneous attribute when the value domain of A_i contains both the clear values (classic values), and the linguistic value. Denoted D_{A_i} is a classic values set of A_i and LD_{A_i} is a linguistic values set of A_i. This time, the inhomogeneous attribute A_i has the value domain $Dom(A_i) = D_{A_i} \cup LD_{A_i}$.

Definitions 2.5. Let $Dom(A_i) = D_{A_i} \cup LD_{A_i}$, v be a semantics quantitative function of $Dom(A_i)$. Function $IC : Dom(A_i) \rightarrow [0, 1]$ is determined:

1. If $LD_{A_i} = \emptyset$ and $D_{A_i} \neq \emptyset$, $\forall \omega \in Dom(A_i)$ we have $IC(\omega) = 1 - \frac{\psi_{\max} - \omega}{\psi_{\max} - \psi_{\min}}$ with $Dom(A_i) = [\psi_{min}, \psi_{max}]$ is a classic value domain of A_i.

2. If $D_{A_i} \neq \emptyset$, $LD_{A_i} \neq \emptyset$, $\forall \omega \in Dom(A_i)$, we have $IC(\omega) = \{\omega \times v(\psi_{maxLV})\}/\psi_{max}$, with $LD_{A_i} = [\psi_{minLV}, \psi_{maxLV}]$ is a linguistic value domain of A_i.

Thus, if we choose the parameters W and fuzziness measure for hedges so that $v(\psi_{maxLV}) \approx 1.0$ then $(\{\omega \times v(\psi_{maxLV})\}/\psi_{max}) \approx 1 - \frac{\psi_{\max} - \omega}{\psi_{\max} - \psi_{\min}}$.

Proposition 2.6. With any inhomogeneous attribute A_i we can homogenize all classic values D_{A_i} and linguistic values LD_{A_i} of A_i to the number value belonging to [0, 1], from that it can transform correspondingly to linguistic value or classic value.

2.4.3. A quantitative way of outlier linguistic valuein the training sample set

Definitions 2.5. Let inhomogeneous attribute $A_i \in D$ we have $Dom(A_i) = D_{A_i} \cup LD_{A_i}$, $D_{A_i} = [\psi_{min}, \psi_{max}]$, $LD_{A_i} = [\psi_{minLV}, \psi_{maxLV}]$. If $x \in LD_{A_i}$ but $v(x) < IC(\psi_{min})$ or $v(x) > IC(\psi_{max})$ then x is called the outlier linguistic value.

Quantitative algorithm for outlier linguistic values

Input: *Inhomogeneous properties contains the outlier linguistic values* A_i
Output: *Homogeneous properties* A_i
Algorithm description:
Separating the alien value out of A, be A'_i *;*
Performing the A'_i *values for uniformity according to the way which a section 2.4.2;*
Compare $\psi_{Outlier}$ *with* ψ_{Max} *and* ψ_{Min} *of* A'_i*. Performing again the partition in* [0, 1]*;*
If $\psi_{Outlier} < \psi_{MinLV}$ then
Begin *Divide* $[0, v(\psi_{MinLV})]$ *into* $[0, v(\psi_{Outlier})]$ *and* $[v(\psi_{Outlier}), v(\psi_{MinLV})]$;
$fm(h_{Outlier}) \sim fm(h_{MinLV}) \times I(\psi_{MinLV})$; $fm(h_{MinLV}) = fm(h_{MinLV}) - fm(h_{Outlier})$;
End;
If $\psi_{Outlier} > \psi_{MaxLV}$ then
Begin *Devide* $[v(\psi_{MaxLV}), 1]$ *into* $[v(\psi_{MaxLV}), v(\psi_{outlier})]$ *and* $[v(\psi_{Outlier}), 1]$;
$fm(h_{Outlier}) \sim fm(h_{MaxLV}) \times I(\psi_{MaxLV})$;$fm(h_{MaxLV}) = fm(h_{MaxLV}) - fm(h_{Outlier})$;
End;
Based on $IC(\omega)$ *of* A'_i*, calculate again* $IC(\omega)$ *for* A_i *;Homogeneous for* A_i.

2.4.4. Fuzzy decision tree algorithm FMixC4.5 based on fuzzy point matching

Algorithm FMixC4.5

Input: Tr*anning set D has n samples, m prediction attributes and decisive attributes Y.*
Output: *Decision Tree S.*
Algorithm description:

Select a typical sample (D);
If (*training set without fuzzy attribute*) then *Call algorithm MixC4.5*;
Else Begin
For each (fuzzy attribute *X* in *D*) do
Begin
Building hedge algebra $\underline{X}_k$ *corresponding to fuzzu attribute* X
Testing and spilting outliers;
Transfer X's number values and linguistic values into interval values $\subset$[0, 1];
Handling the outliers
End;
Call algorithm MixC4.5;
End;

The complexity of FMixC4.5 is $O(m \times n^2 \times logn)$.

2.4.5. Experimental implementation and evaluation of the FMixC4.5 algorithm

Table 2.8. A comparison of the results with the 5000 training samples of the FMixC4.5 on the database with fuzzy attribute Mushroom

Algorithm	Time training	The number of samples to check for the predictive accuracy				
		100	**500**	**1000**	**1500**	**2000**
C4.5	18.9	0.570	0.512	0.548	0.662	0.700
MixC4.5	50.2	0.588	0.546	0.548	0.662	0.700
FMixC4.5	58.2	0.710	0.722	0.726	0.779	0.772

Table 2.9. The test time comparison table with 2000 samples of the FMixC4.5 on the database with fuzzy attribute Mushroom

Algorithm	The number of test samples and the predicted execution time (s)				
	100	**500**	**1000**	**1500**	**2000**
C4.5	0.2	0.7	1.6	2.1	2.9
MixC4.5	0.2	0.8	1.7	2.2	3.0
FMixC4.5	**0.4**	**1.0**	**1.9**	**2.8**	**3.8**

- **Cost of Time:** Although with the same complexity level but MixC4.5 always performs faster than FMixC4.5 during the training and prediction period. MixC4.5 ignores the fuzzy values in the sample set so that it does not take time to process, and it has to undergo the construction of the hedge algebras for fuzzy fields to homogenise the fuzzy values and handle the outliers, so FMixC4.5 is slower than C4.5 and MixC4.5.
- **The prediction result:** Because MixC4.5 ignores fuzzy values

in the sample set, only clear values are concerned, it loses data in fuzzy fields, so the predicted results are not high because it cannot effectively predict for the cases containing fuzzy values. Homogenizing the sample set for the training sample set containing precise and imprecise data, so the result tree trained by FMixC4.5 is better, the prediction result is higher if we use C4.5 and MixC4.5.

2.5. Summary

In order to overcome the limitations of traditional decision tree learning algorithms, this chapter of the thesis focuses on:

1. Analyzing the correlation between tree-based learning algorithms and analyzing the influence of the training sample set on the result tree, presented a method for selecting the typical training sample set support for the training process and proposed algorithm MixC4.5 for learning process.

2. Analyzing and introducing the concepts of heterogeneous sets, the outlier, and building an algorithm that can homogenise the attributes containing these values.

3. Building algorithm FMixC4.5 to support for the decision tree learning process on the inhomogeneous sample set. The matched experimental implementation results showed the predictability of MixC4.5, FmixC4.5 more effective than other traditional algorithms.

Chapter 3.
FUZZY DECISION TREE TRAINING METHODS FOR DATA CLASSIFICATION PROBLEM BASED ON FUZZINESS INTERVALS MATCHING

3.1. Introduction

For the purpose of constructing a decision tree model *S* with high effective for the classification process, i.e. $f_h(S) \rightarrow max$ on the training set *D*, Chapter 2 of this thesis focused on solving the constraints of traditional learning methods by introducing the MixC4.5 and FMixC4.5 learning algorithms. However, due to the homogenizing process of the linguistic value LD_{A_i} and the numerical value of D_{A_i} of the fuzzy attribute A_i of the values in [0, 1] causes the errors. There are many approximate classic values reduced to one point in [0, 1], so the predicted result of FMixC4.5 has not really met the expectations.

In addition, with the goal set at (1.10), the goal function $f_h(S) \rightarrow max$ also implies the flexibility in predict process, which has

predictability for many different cases. In addition, the division at the fuzzy attributes in the result tree model according to the dividing points makes it difficult in the case of predictions of value intervals with alternant value domains between the two branches of the tree.

3.2. The fuzziness interval values matching method of the fuzzy attribute

3.2.1. Building an interval values matching method based on the hedge algebra

Definition 3.3: Let $[a_1, b_1]$ and $[a_2, b_2]$ be two different precise intervals corressponding to the fuzzines 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 $[a_2, b_2]$ 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.* $I_{b_2} > I_{b_1}$);

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 3.1. 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.

3.2.2. The fuzziness interval determining method when do not determine *Min, Max* value of fuzzy attributes

Definition 3.4. For homogeneous attribute A_i, we have $Dom(A_i) = D_{A_i} \cup LD_{A_i}$, $D_{A_i} = [\psi_1, \psi_2]$ and $LD_{A_i} = [\psi_{minLV}, \psi_{maxLV}]$. A_i is called an inhomogeneous fuzzy attribute, do not determine *Min-Max* when $\psi_{minLV} < \psi_{LV1}, \psi_{LV2} < \psi_{maxLV}$ where $v(\psi_{LV1}) = IC(\psi_1)$ and $v(\psi_{LV2}) = IC(\psi_2)$.

Algorithm to determine fuzziness intevals for heterogeneous attributes, unknown Min-Max

Input: inhomogeneous attribute, unknown *Min-Max* A_i

Output: Attribute with homogenized domain by fuzziness inteval A_i

Algorithm description:

Build *hedge algebras in* $[\psi_1, \psi_2]$; Compute $IC(\omega_i)$ corresponding to the values in $[\psi_1, \psi_2]$;

For each ($v(\psi_{LV_i}) \notin [IC(\psi_1), IC(\psi_2)]$) do

Begin

If $v(\psi_{LV_i}) < IC(\psi_1)$ then Begin

Partition$[0, v(\psi_1)]$ *into* $[0, v(\psi_i)]$ and $[v(\psi_i), v(\psi_1)]$;

Compute $fm(h_i) \sim fm(h_1) \times I(\psi_1)$ and $fm(h_1) = fm(h_1) - fm(h_i)$;

Compute $\omega_i = v(\psi_1) \times \frac{IC(\omega_1)}{IC(\omega_i)}$ and $IC(\omega_i)$; Assign position ψ_i to position ψ_1;

End;

If $\nu(\psi_{LV_i}) > IC(\psi_2)$ then Begin

Partition[$\nu(\psi_2)$, 1] *into* [$\nu(\psi_2)$, $\nu(\psi_i)$] and [$\nu(\psi_i)$, 1];

Compute $fm(h_i) \sim fm(h_2) \times I(\psi_2)$ and $fm(h_2) = fm(h_2) - fm(h_i)$;

Compute $\omega_i = \nu(\psi_2) \times \frac{IC(\omega_2)}{IC(\omega_i)}$ and $IC(\omega_i)$; Assign position ψ_i to position ψ_2;

End;

End;

3.3. Learning classification by the fuzzy decision tree based on fuzziness interval matching

3.3.1. Fuzzy decision tree learning algorithm HAC4.5 based on fuzziness interval matching

The Information gain of fuzziness intervals at the fuzzy attribute

With fuzzy attribute A_i quantified according to the fuzziness interval without losing the generality and there are *k*different intervals with post-preorder relations:

$$[I_{a_1}, I_{b_1}] < [I_{a_2}, I_{b_2}] < \ ... < [I_{a_k}, I_{b_k}] \qquad (3.1)$$

We have *k* thresholds computed*:* $Th_i^{HA} = [I_{a_i}, I_{b_i}]$, ($1 \le i < k$). At each threshold Th_i^{HA} of the selected fuzziness interval $[I_{a_i}, I_{b_i}]$ the set of data *D* of this remaining node are divided into two sets:

$$D_1 = \{\forall [I_{a_j}, I_{b_j}] : [I_{a_j}, I_{b_j}] \le Th_i^{HA})\} \qquad (3.2)$$

$$D_2 = \{\forall [I_{a_j}, I_{b_j}] : [I_{a_j}, I_{b_j}] > Th_i^{HA})\} \qquad (3.3)$$

Then, we have:

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

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

$$GainRatio^{HA}(D, Th_i^{HA}) = \frac{Gain^{HA}(D, Th_i^{HA})}{SplitInfo^{HA}(D, Th_i^{HA})}$$

Based on computing the information gain ratio of thresholds, we will select a threshold which has the most information.

Algorithm HAC4.5

Input: Training data set *D*.

Output**:** Fuzzy decision tree*S*.

Algorithm description:

For each *(fuzzy attribute X in D)* do

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 has GainRatio or *GainRatio*^{HA} *is the biggest;*
L.Label = Attribute name X;
If (*L is fuzzy attribute*) then
Begin
T = Threshold has GainRatio^{HA}*is the biggest;*
Add label T into S;
$S_1 = \{I_{x_i} : I_{x_i} \subseteq L, I_{x_i} \leq T\}$; $S_2 = \{I_{x_i} : I_{x_i} \subseteq L, I_{x_i} > T\}$;
Creating two little buttons for current button which correspond with S_1 *and* S_2 ;
Marking L button;
End
Else
If (*L is continuous attribute*) then
Begin
T = Threshold has GainRatio is the biggest;
$S_1 = \{x_i : x_i \in Dom(L), x_i <= T\}$; $S_2 = \{x_i : x_i \in Dom(L), x_i > T\}$;
Creating two little buttons for current button which correspond with S_1 *and* S_2 ;
Marking L button;
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 $S_i = \{x_j : x_j \in Dom(L), x_j = x_i\}$;
Creating a little button i for current button and correspond with S_i;
End;
Marking L button;
End;
End;

The complexity of HAC4.5 is $O(m \times n^2 \times log\ n)$.

3.3.2. Experimental implementation and evaluation of HAC4.5 algorithm

Table 3.4. Compare the results with the 20000 training samples of C4.5, FMixC4.5 and HAC4.5 on data containing the fuzzy attribute Adult

Algorithm	Time training	The number of test samples and predictive accuracy				
		1000	2000	3000	4000	5000
C4.5	479.8	0.845	0.857	0.859	0.862	0.857
FMixC4.5	589.1	0.870	0.862	0.874	0.875	0.866
HAC4.5	1863.7	0.923	0.915	0.930	0.950	0.961

Table 3.5. Time matching to test from 1000 to 5000 samples on Adult data

Algorithm	The number of samples tests and the predictive accuracy				
	1000	2000	3000	4000	5000
C4.5	1.4	2.8	4.1	5.5	6.0
FMixC4.5	2.2	4.6	7.1	9.2	11.8
HAC4.5	2.4	4.7	7.2	9.7	12.1

Evaluation of experimental results

Cost of time: Because there is a need for the construction of the hedge algebras for the fuzziness fields and the cost for the conversion of values to the initial interval [0, 1], futhermore, at each loop additional time is necessary for the selection of intervals, the algorithm HAC4.5 is relatively slow compared with other algorithms.

The prediction result: The predition results of HAC4.5 is the best because in the tree training, the imprecise values are processed while the imprecise values remain unchanged, leading to the absence of errors in the partition process. Although HAC4.5 needs more time for training, it is an effective method as the result tree has high predictability. Futhermore, the training process is performed only once while the prediction on the result tree is done for several times, and thus the processing time of HAC4.5 is acceptable.

3.4. Constructing the concept of maximum fuzzinessintervals and method for optimizing the fuzzy decision tree model

3.4.1. The multi-objective problem of fuzzy decision tree

Firstly, we need to reiterate that the objective of the problem mentioned in (1.10) is $f_h(S) \rightarrow max$ and $f_n(S) \rightarrow min$. The studies in Chapter 2 and Section 3.3 of the thesis are a compromise in achieving the goal $f_h(S) \rightarrow max$ and $f_n(S) \rightarrow min$ is not solved.

3.4.2. The concept of the maximum fuzziness interval and how to calculate the maximum fuzziness interval for fuzzy attributes

Definition 3.5. Let $\underline{X} = (X, G, H, \preceq)$ be a hedge algebra,where $\forall x, y \in X$ are in the semantic inheritance relationship and are denoted by $\sim(x, y)$ if $\exists z \in X, x = h_{i_n} .. h_{i_1} z, y = h_{j_m} .. h_{j_1} z$.

Proposition 3.1. $\forall\ x, y \in X$ define two fuzziness intervalsof k and l, respectively $I_k(x)$ và $I_l(y)$, which either do not have inheritance or inheritance if $\exists z \in X, |z| = v, v \preceq min(l, k), I_L(z) \preceq I_L(y), I_R(z) \succeq I_R(y)$, and $I_L(z) \preceq I_L(x), I_R(z) \succeq I_R(x)$ or $I_v(z) \supseteq I_k(x)$ and $I_v(z) \supseteq I_l(y)$, i.e. x, y are

generated from z.

Definition 3.6. Let $\underline{X} = (X, G, H, \preceq)$ be a hedge algebra, with $x, y, z \in X$, $z = \sim(x, y)$. If $\exists z_1 \in X$, $z_1 = \sim(x, y)$ and $len(z) \geq len(z_1)$ then we say that semantically close to x, y most, or fuzziness interval z has maximum length and signed $z = \sim_{max}(x, y)$.

Definition 3.7. Let $\underline{X} = (X, G, H, \preceq)$ be a hedge algebra,with $\forall x, y \in X$ and $\sim(x, y)$. The approximate degree of x and y in the semantic inheritance relationship is $sim(x, y)$ and is defined as:

$$sim(x,y) = \frac{m}{max\,(k,l)}(1 - |v(x) - v(y)|) \quad (3.7)$$

Where $k = len(x)$, $l = len(y)$ and $m = len(z)$ with $z = \sim_{max}(x, y)$.

Proposition 3.2. Let $\underline{X} = (X, G, H, \preceq)$ be a hedge algebra,with $\forall x, y \in X$, we have the properties of the degree of proximity of the terms as follows:

1. Function $sim(x, y)$ is symmetric, i.e. $sim(x, y) = sim(y, x)$
2. x, y does not have the semantic inheritance relationship $\Leftrightarrow sim(x,y) = 0$
3. $sim(x, y) = 1 \Leftrightarrow x = y$,
4. $\forall x, y, z \in X_k, x \leq y \leq z \Rightarrow sim(x, z) \leq sim(x, y)$, $sim(x, z) \leq sim(y, z)$.

Definition 3.8. Definition of contiguity of fuzziness intervals. Let $\underline{X} = (X, G, H, \preceq)$ be a hedge algebra, the two fuzziness intervals $I(x)$ and $I(y)$ are called contiguous if they have a common point, i.e. $I_L(x) = I_R(y)$ or $I_R(x) = I_L(y)$.

The algorithm points out the maximum fuzziness interval from two given fuzziness intervals

Input: *Hedge algebras* $\underline{X} = (X, G, H, \preceq)$ *and* $x, y \in X$.
Output: $z \in X$, $z = \sim_{max}(x, y)$.
Algorithm description:

$k = len(x)$; $l = len(y)$; $v = min(k, l)$;
While $v > 0$ do
  If $\exists z \in X, |z| = v$ and $I_k(x) \subseteq I_v(z)$ and $I_l(y) \subseteq I_v(z)$ then return $I_v(z)$
  Else $v = v - 1$;
Return *NULL;*

3.4.3. Fuzzy decision tree algorthim HAC4.5* based on maximum fuzziness intervals

Because the fuzzy attribute A of the training set was already partitioned by the fuzziness inteval is a sub-interval of [0, 1] and its data domain is a linearly ordered set according to the pre-order, post-order relations. So their intervals will be on the left or right. So with the

two fuzziness intevals x and y if they share the same predictive class, we can use the fuzziness interval $z = \sim_{max}(x, y)$ without changing the semantics of x and y in the classification learning process. The use of the z join instead of x and y is done for all fuzziness intevals of the fuzzy attribute A.

Algorithm HAC4.5*

Input: Training data set *D*.
Output: Fuzzy decision tree *S*.
Algorithm description:

```
For each (fuzzy attribute X in D)
Begin
    Built a hedge algebra X_k corresponding with fuzzy attribute X;
    Transform number values and linguistic values of X into intervals ⊆ [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
        If (L is fuzzy attribute) then
        Begin   For each (fuzziness interval x of attribute L)
                    For each (fuzziness intervaly of attribute L mà y ≠ x)
                        Findand replace x with z = ~max(x, y);
        End;
        X is attibute has GainRatio or GainRatio^HA is the biggest;
        If (L is fuzzy attribute) then
        Begin
            T = Threshold has GainRatio^HA is the biggest;Add label T into S;
            S1= {I_xi |I_xi ⊆ L, I_xi <T}; S2= {I_xi | I_xi ⊆ L, I_xi > T};
        Creating two little buttons for current button which correspond with S1 and S2 ;
            Marking L button;
        End
        Else
        If (L is continuous attribute) then
            Begin
            T = Threshold has GainRatio is the biggest;
            S1= {xi :xi ∈Dom(L), xi <= T}; S2= {xi :xi ∈Dom(L), xi> T};
            Creating two little buttons for current button which correspond with S1 and S2 ;
            Marking L button;
            End
        End
        Else { L is discrete attribute }
        Begin
            P = {xi : xi ∈ K, xi single};
            For (each xi ∈ P)do
```

Begin

$S_i = \{x_j : x_j \in Dom(L), x_j = x_i\}$;

Creating a little button i for current button and correspond with S_i;

End;

Marking L button;

End;

End;

With *m* is the number of attributes, n is the number of the training set, the complexity of HAC4.5* is $O(m \times n^3 \times log\ n)$. The correctness and stopping of the algorithm is derived from the correctness of C4.5 and how the fuzzy values are matched.

3.4.4. Experimental implemetation and evaluation of the algorithm HAC4.5*

Table 3.6. Training results in data Adult

Algorithm	Training time(s)	Nodes in the tree
C4.5	479.8	682
HAC4.5	1863.7	1873
HAC4.5*	2610.8	1624

Table 3.7.Checking rate in data Adult

Checking sample / Algorithm	1000	2000	3000	4000	5000
C4.5	84.5%	85.7%	85.9%	86.2%	85.7%
HAC4.5	92.3%	91.5%	93.0%	95.0%	96.1%
HAC4.5*	92.8%	91.6%	93.2%	95.1%	96.3%

Comparison of experimental results FMixC4.5, HAC4.5 and HAC4.5 * with some results of other approaches

Training costs: HAC4.5 * at each loop additional time is necessary for searching the maximum fuzziness intervals for fuzzy value domain of the correlative fuzzy attribute, so HAC4.5* is the slowest compared with other algorithms.

Predictable results: The result of HAC4.5* is the best because in the tree training process, we found out the best partition points at the fuzzy attributes, so the result tree has some errors. Moreover, finding the maximum fuzzy intervals and joining the fuzzy values with fuzzy attributes that reducing the corelative fuzzy attributes, the nodes of the tree also reduces, so the result tree is the best. This is suitable for the goal of Section 3.4.1.

Futhermore, algorithms matching proposed some algorithms to the

existing algorithms, Figure 3.8 shown that using hedge algebra for fuzzy classification problem is effective.

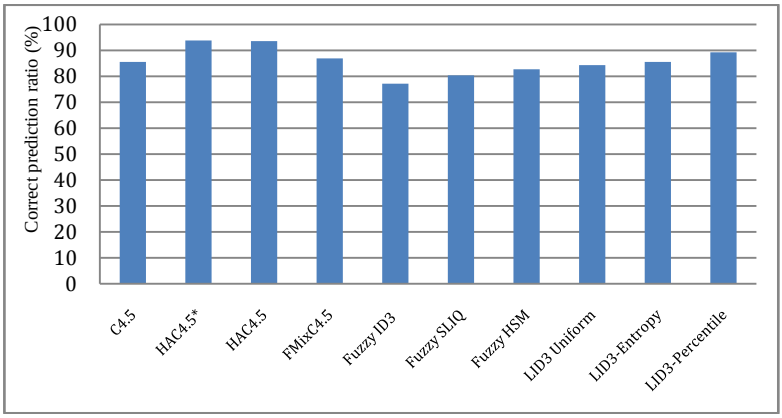


Figure 3.8. Comparison of the predict rate of althgrithm FMixC4.5, HAC4.5 và HAC4.5* with the other approaching

3.5. Summary

This chapter focuses on studying the learning process of fuzzy decision tree to achieve two goals: $f_h(S) \rightarrow max$ and $f_n(S) \rightarrow min$.

1. Study the correlation of fuzziness intervals, matching methods based on fuzziness intervals, and build the classification algorithm based on fuzziness interval HAC4.5.

2. Study and show that the Min-Max domain of the fuzzy attribute does not always exist in the training set. Based on the nature of the hedge algebra, the thesis built a method to quantify the values of the inhomogeneous attributes, unknown *Min-Max* of the training set.

3. The thesis proposed the concept of the maximum fuzziness intervals, designed the HAC4.5* algorithm to achieve the objectives.

CONCLUSION

The main result of this thesis is to study, propose models and methods for decision tree training in order to obtain result trees that are effective in classifying and simple for users to understand. The main contents of the thesis were as follow:

1. Proposed a model of decision tree training from a practical

training sample set and a method to select a specific training sample set for the training process. Analyzed, introduced the concept of inhomogeneous sample sets, outliers, built the algorithms that can homogenize the attributes containing these values.

2. Proposed the algorithm to build the tree MixC4.5 based on the synthesis of the advantages and disadvantages of traditional algorithms CART, C4.5, SLIQ, SPRINT. Pointed out the limitations of the FDT and FID3 algorithms for the fuzzy decision tree learning, the thesis proposed the FMixC4.5 algorithm for learning the decision tree on the inhomogeneous sample set. Both the MixC4.5 and FMixC4.5 algorithms were evaluated experimentally on the Northwind and Mushroom databases, and the results were better than the traditional algorithms C4.5, SLIQ, and SPRINT.

3. Proposed the matching method based on fuzziness intervals and built classification algorithm based on fuzziness intervals HAC4.5. Built the quantative method for the values of inhomogeneous attributes, unknown Min-Max of the training set.

4. The thesis presented the concept of maximum fuziness interval which is used to design decision tree algorithm based on the maximum fuzziness interval HAC4.5* in order to achieve the effective of classification process, simple for the users. The results of HAC4.5, HAC4.5* are analysed, evaluated experimentally on database Mushroom, Adult and the results got high predictability and more nodes on the training tree.

However, in selecting the parameters for the building hegde algebra to quantify the linguistic value on the training sample set, the thesis is using the expert's knowledge to identify parameters without studying to give out a complete method.

Development direction of the thesis:

- Studying aims at providing an appropriate method for selecting parameters for hegde algebra of the training set.

- Extending the decision tree learning method based on fuzziness inteval without the limitation of hedge algebra while buiding hedge algebra for homogenizing the fuzzy attributes.

- Based on the application model in the classification problem, continued to develop models to apply to some other problems in the field of data mining.

REFERENCE

CT1. Le Van Tuong Lan, Nguyen Mau Han, Nguyen Cong Hao, *An algorithm for building decision tree in data classification problem*, Journal of Science, Hue University, Vol. 81, Num. 3, pages 71 - 84, 2013.

CT2. Le Van Tuong Lan, Nguyen Mau Han, Nguyen Cong Hao, *An Approach for choosing a traning set to build a decision tree based on hedge algebra*, Proceedings of 6th National conference on Fundamental and Applied International Technology Research (FAIR), pages 251 - 258, 2013.

CT3. Le Van Tuong Lan, Nguyen Mau Han, Nguyen Cong Hao, *A method for handling outliers in training data set to build a decision tree based on hedge algebra*, Research, Development and Application on Information & Communications Technology, Journal of Information, Science and Technology, Ministry of Information and Communications, Vol. 2, No. 14, pages 55 - 63, 2015.

CT4. Lan L. V., Han N. M., Hao N. C., *A Novel Method to Build a Fuzzy Decision Tree Based On Hedge Algebras*, International Journal of Research in Engineering and Science (IJRES), Volume 4 Issue 4, pages 16 - 24, 2016.

CT5. Le Van Tuong Lan, Nguyen Mau Han, Nguyen Cong Hao, *Algorithm to build fuzzy decision tree for data classification problem based on fuzziness intervals matching*, Journal of Computer Science and Cybernetics, V.32, N.4, DOI 10.15625/1813-9663/30/4/8801, pages 367 - 380, 2016.

CT6.Le Van Tuong Lan, Nguyen Mau Han, Nguyen Cong Hao, *Fuzzy decision tree model for data classification problem*, Journal of Sciences and Technology, Hue University College of Sciences, Vol. 8, No. 1, pages 19 - 34, 2017.

CT7. Le Van Tuong Lan, Nguyen Mau Han, Nguyen Cong Hao, *Optimal the learning fuzzy decision tree for data classification problem based on an approach of maximum fuzziness intervals*, Research, Development and Application on Information & Communications Technology, Journal of Information, Science and Technology, Ministry of Information and Communications, Vol. 2, No. 18 (38), pages 42 - 50, 2017.