

Soft Discretization to Enhance the Continuous Decision Tree Induction*

Yonghong Peng¹ Peter A Flach

Department of Computer Science, University of Bristol, Bristol, BS8 1UB, UK.
{yh.peng, peter.flach}@bristol.ac.uk

Abstract. Decision tree induction has been widely used to generate classifiers from training data through a process of recursively splitting the data space. In the case of training on continuous-valued data, the associated attributes must be discretized in advance or during the learning process. The commonly used method is to partition the attribute range into two or several intervals using a single or a set of cut points. One inherent disadvantage in these methods is that the use of sharp (crisp) cut points makes the induced decision trees sensitive to noise. To overcome this problem this paper presents an alternative method, called soft discretization, based on fuzzy set theory. As opposed to a classical decision tree, which gives only one class as the end result, the soft discretization based decision tree associates a set of possibilities to several or all classes for an unknown object. As a result, even if uncertainties existed in the object, the decision tree would not give a completely wrong result, but a set of possibility values. This approach has been successfully applied to an industrial problem to monitor a typical machining process. Experimental results showed that, by using soft discretization, better classification accuracy has been obtained in both training and testing than classical decision tree, which suggest that the robustness of decision trees could be improved by means of soft discretization.

1. Introduction

Decision trees have been successful in classifications, and various algorithms have been proposed for the generation of decision trees [1,2,3]. Decision tree induction methods such as ID3, C4.5, and CART generate a tree structure through recursively partitioning the attribute space until the whole decision space is completely partitioned into a set of non-overlapping subspaces. The original decision tree is restricted to attributes that take on a discrete set of values. If continuous attributes are involved, they must be discretized appropriately. Two typical methods have been used for this purpose, one of which is to partition the attribute range into two intervals using a threshold [4], while another is to partition the attribute domain into several

* Supported by Esprit METAL project (26.357), and National Natural Science Foundation of China (59905008).

¹ Research Fellow of University of Bristol from the South China University of Technology.

intervals using a set of cut points [5]. This paper concentrates primarily on the first approach to discretize the attribute into two intervals, as it has been used in many data mining tools and software.

In both methods above, the cut points used in classical decision trees are usually crisp. Applications showed that these approaches could only work well for disjoint classes that can be separated with clearly defined boundaries. However, due to the existence of vague and imprecise information in real-world problems, the class boundaries may not be defined clearly. In this case, the decision tree may produce high misclassification rates in testing even if they perform well in training. To overcome this drawback, several approaches have been proposed, including probability based approaches [6,7,8], and possibility based methods [9,10]. What is the major cause to this issue? Given an unknown object, only one branch of tree (one rule) is fired at a time in classical decision tree, which may lead to opposite following branch when the input is close to the cut point, and thus generate a wrong result. This drawback could be overcome by firing multi-branches but with various certainty degrees. This could be implemented by means of fuzzy set theory, leading to fuzzy decision trees. In fuzzy decision trees the fuzzy reasoning process allows two or more rules to be simultaneously validated with gradual certainty and the end result will be the outcome of combining several results.

The induction of fuzzy decision trees follows the same steps as that of a classical decision tree with modified induction criteria [11]. Yuan proposed a novel criterion based on the measurement of cognitive uncertainty [12], Peng proposed an alternative criterion based on fuzzy mutual entropy in possibility domain [13]. In these approaches, the continuous attributes are needed to be partitioned into several fuzzy sets prior to the tree induction, heuristically based on expert experiences and the data characteristics. In this paper, the attribute domain is discretized in the process of tree induction by using two fuzzy sets. These two fuzzy sets form a soft discretization. Furthermore, the soft discretization based induction method, including the criteria for attribute discretization and selection, as well as the stopping criterion of tree induction, is presented in detail in this paper. The effectiveness of the proposed method has been verified in an industrial application to monitor a typical machining process. The experimental results showed that, comparing to the classical decision tree, higher classification accuracy was obtained in testing.

This paper is organised as following. Section 2 introduces fuzzy decision trees and fuzzy classification. In section 3, a brief review of induction of decision trees from continuous-valued data is given. The concept of soft discretization and the associated continuous decision tree induction is presented in section 4. In Section 5, a typical application is introduced to illustrate the effectiveness of using soft discretization.

2. Fuzzy Decision Tree and Fuzzy Classification

The potential of fuzzy decision trees in improving the robustness and generalization in classification is due to the use of fuzzy reasoning. Fig.1 illustrates the difference between classical and fuzzy decision trees. Fig.1(a) shows a classical decision tree based on crisp discretization, while the tree in Fig.1(b) uses soft discretization. In both

decision trees, each path from the root node to a leaf node establishes a classification rule. Using crisp discretization, the decision space is partitioned into a set of non-overlapping subspaces, as shown on the right of Fig.1(a), in which each data point is assigned to a single class. In contrast, a fuzzy decision tree gives results within $[0,1]$, as the possibility degree of an object matching the class, as shown on the right of Fig.1(b). Fuzzy decision trees provide a more robust way to avoid misclassification. For example, given an object $(x_1=81, x_2=25)$, the conclusion reached by the classical decision tree is class c_3 . However, if the sampled data point has moved to a new point with a small change of value due to noise, for example $(x_1=79, x_2=25)$, then the classical decision tree would give a wrong result: class c_2 . In contrast, the fuzzy decision tree gives a result about the possibilities that the object belongs to classes c_1, c_2, c_3 , for instant $\pi_1=0, \pi_2=0.52, \pi_3=0.48$ respectively. According to these results, the human users could make their final decision or perform further investigation, or a high-level meta-learner can be designed to learn further. As a result, the rate of misclassification could be reduced.

Each branch of a fuzzy decision tree, from the root to the leaf, forms a decision rule, which can be represented in the format of “if x_1 is A_1 and x_i is A_i and x_m is A_m then class is c_j ”, in which ‘ x_1 is A_1 ’.... ‘ x_i is A_i ’.... and ‘ x_m is A_m ’ are fuzzy decisions at nodes and the associating branches, and c_j is the class in the leaf. One example of rule obtained in Fig.1(b) is “if x_2 is B_2 and x_1 is A_1 then class is c_2 ”.

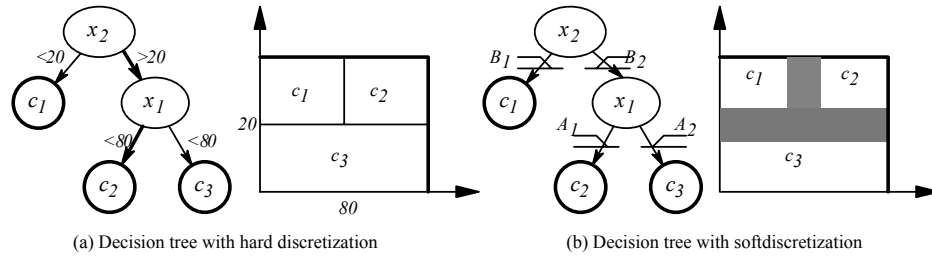


Fig. 1. Decision tree and the decision space

The classification for a given unknown object is obtained from the matching degrees of the object to each node from root to leaf. In the above example, the possibility of an object belonging to class c_2 is calculated by:

$$\pi_2 = \otimes[B_2(x_2), A_1(x_1)], \quad (1)$$

where the circled plus \otimes is the fuzzy product operation (the minimum or weighted average is usually employed), and $B_2(x_2)$ and $A_1(x_1)$ are the membership degree of x_2 to B_2 and x_1 to A_1 respectively. In the same way, the possibility of the object belonging to each class can be calculated, $\{\pi_i\}_{i=1-k}$. If more than one leaf are associated with a same class c_i , say, the value of $\pi_i = \oplus(\pi_j)$ will be considered as the

possibility of the corresponding class, where the maximum operation is used as the fuzzy sum operation \oplus . In the end, if one possibility value, such as π_K , is much higher than others, that is $\pi_K \gg \pi_{i \neq K}$, then the class c_K will be assigned as the class of the object, otherwise the decision tree predicts a distribution over all the classes.

3. Inducing Decision Trees from Continuous Data

In the induction of decision trees from continuous-valued data, a suitable threshold T , which discretizes the continuous attribute A into two intervals $A_1 = [\min(A), T]$ and $A_2 = (T, \max(A)]$, is determined based on the classification information gain generated by the corresponding discretization. Given a threshold, the test $A \leq T$ is assigned to the left branch of the decision node while $A > T$ is assigned to the right branch. Assuming we are to select an attribute for a node having a set S of N examples, these examples are sorted according to the values of the continuous attribute A ; and an ordered sequence of distinct values a_1, a_2, \dots, a_N is formed. Every pair of adjacent data points suggest a potential threshold $T = (a_i + a_{i+1})/2$ to create a cut point and generate a corresponding partition of A . Fayyad had proved that only the class boundary points could be the cut points to obtain the maximum information in classification, which implies if a_i and a_{i+1} belong to the same class, a cut point between them can not lead to a partition that has maximum information gain [4]. Therefore, we can generate a smaller set of candidate cut points from the class boundary points. Let there be k classes c_1, c_2, \dots, c_k , and let $p(c_j, S)$ be the proportion of examples in S that belong to class c_j . The residual uncertainty in classification is expressed as the class entropy:

$$E(S) = -\sum_{j=1}^k p(c_j, S) \log(p(c_j, S)), \quad (2)$$

After the set of example S is partitioned into two subsets S_1 and S_2 by a threshold T , the class information entropy is expressed as the weighted average of their resulting class entropy:

$$E(A, T; S) = \frac{N_1}{N} E(S_1) + \frac{N_2}{N} E(S_2), \quad (3)$$

$$E(S_1) = -\sum_{j=1}^k p(c_j, S_1) \log(p(c_j, S_1)), \quad (4)$$

$$E(S_2) = -\sum_{j=1}^k p(c_j, S_2) \log(p(c_j, S_2)), \quad (5)$$

where $N_1 = |S_1|$, $N_2 = |S_2|$ and $N = |S|$ are the number of examples in S_1 , S_2 , and S , and $p(c_j, S_1)$ and $p(c_j, S_2)$ are the proportion of examples of class c_j in S_1 and S_2 respectively. The cut point for which $E(A, T_A; S)$ is minimal among all the candidate cut points of attribute A is used; and the attribute A_j , for which the $E(A_j, T_A^j; S)$ is minimum, or the information gain $E(S) - E(A_j, T_A^j; S)$ is maximum, will be then selected to generate two child nodes. In each child node, discretization and attribute

selection are performed again based on the partitioned examples. The above process is repeated recursively until a stopping criterion is matched.

4. Continuous Decision Tree Induction based on Soft Discretization

Soft discretization could be viewed as an extension of hard discretization, and the classical information measures defined in the probability domain have been extended to new definitions in the *possibility domain* based on fuzzy set theory [13]. A crisp set A_c is expressed with a sharp characterization function $A_c(a) : \Omega \rightarrow \{0,1\}; a \in \Omega$, alternatively a *fuzzy set* A is characterized with a *membership function* $A(a) : \Omega \rightarrow [0,1]; a \in \Omega$. The membership $A(a)$ is called the possibility of A to take a value $a \in \Omega$ [14]. The *probability of fuzzy set* A is defined, according to Zadeh [15], by $P_F(A) = \int_{\Omega} A(a) dP$, where dP is a probability measure on Ω , and the subscript F is used to denote the associated *fuzzy* terms. Specially, if A is defined on discrete domain $\Omega = \{a_1, \dots, a_i, \dots, a_m\}$, and the probability of $P(a_i) = p_i$ then its *probability* is $P_F(A) = \sum_{i=1}^m A(a_i) p_i$.

Let $Q = \{A_1, \dots, A_k\}$ be a family of fuzzy sets on Ω . Q is called a *fuzzy partition* of Ω [16] when:

$$\sum_{\gamma=1}^k A_{\gamma}(a) = 1, \quad \forall a \in \Omega. \quad (6)$$

A hard discretization, as shown in Fig.2(a), is defined with a threshold, which generates the boundary between two crisp sets. Alternatively, a soft discretization is defined by a fuzzy set pair which forms a fuzzy partition (Eq.(6)), as shown in Fig.2(b). In contrast to the classical method of non-overlapping partitioning, the soft discretization is overlapped. The soft discretization is defined with three parameters/functions, one is the cross point T , the other two are the membership functions of the fuzzy set pair A_1 and A_2 : $A_1(a) + A_2(a) = 1$. The cross point T , i.e. the localization of soft discretization, is determined based on whether it can maximize the information gain in classification, and the membership functions of the fuzzy set pair are determined according to the characteristics of attribute data, such as the uncertainty of the associated attribute. Usually, wide overlapping is used for a high uncertain attribute, for example, we can use the average distance of data points as the overlapping width.

The fuzzy class entropy in S is defined as $E_F(S) = \sum_{j=1}^k p(c_j, S) \log p(c_j, S)$, where $p(c_j, S) = \sum_{a_i \in c_j} (A_1(a_i) + A_2(a_i))$ is the fuzzy proportion of examples in S . After soft discretization, the class information entropy is calculated with the probability of fuzzy partition, as:

$$E_F(A, T; S) = \frac{N_F^{S_1}}{N_F^S} E_F(S_1) + \frac{N_F^{S_2}}{N_F^S} E_F(S_2), \quad (7)$$

$$E_F(S_1) = -\sum_{j=1}^k p(c_j, S_1) \log p(c_j, S_1), \quad (8)$$

$$E_F(S_2) = -\sum_{j=1}^k p(c_j, S_2) \log p(c_j, S_2), \quad (9)$$

$$p(c_j, S_k) = N_F^{S_k c_j} / N_F^{S_k}, k=1,2, \quad (10)$$

where $N_F^S = \sum_{i=1}^{|S|} (A_1(a_i) + A_2(a_i))$, $N_F^{S_1} = \sum_{i=1}^{|S_1|} A_1(a_i)$, $N_F^{S_2} = \sum_{i=1}^{|S_2|} A_2(a_i)$, and $N_F^{S_k c_j} = \sum_{a_i \in c_j} A_k(a_i)$, ($k=1,2$). Similar to classical decision induction, the information gain, $E_F(S) - E_F(A, T; S)$, is used as the criterion to generate the best discretization for the corresponding attribute.

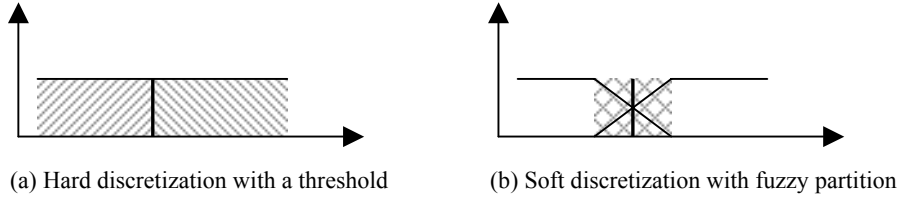


Fig. 2. Hard discretization and soft discretization

The details include eight steps, as following:

Step.1: Assume current node contains N examples, denoted as S , sorting the examples according to the values of A generates a sequence of ordered values a_1, a_2, \dots, a_N ;

Step.2: Generate candidate cut points $T=(a_i+a_{i+1})/2$ using the class boundary points;

Step.3: Fuzzify the cut points to generate candidate soft discretizations, by using the fuzzy set pair, A_1 and A_2 , which form a fuzzy partition on the cut point;

Step.4: Evaluate each candidate soft discretization using Eq.(5);

Step.5: Select the soft discretization having a minimum value of E_F , to attribute A .

Step.6: Repeat Step.1-Step.5 to generate the soft discretization for other attributes;

Step.7: Select the attribute A_j , whose discretization has the minimum value of $E_F(A, T; S)$ to generate two child branches and nodes;

Step.8: Calculate the truth level for each of two branches:

$$\eta_1 = \frac{N_F^{S_1}}{N_F^S}, \quad \eta_2 = \frac{N_F^{S_2}}{N_F^S}. \quad (11)$$

If $\eta_1 \leq \alpha$ or $\eta_2 \leq \alpha$, then delete the corresponding branches. If $\eta_1 > \alpha$ or $\eta_2 > \alpha$, then calculate the truth level of each branch belonging to the j th class:

$$\mu_{1j} = \frac{\sum_{a_i \in c_j} A_1(a_i)}{N_F^{S_1}}, \quad \mu_{2j} = \frac{\sum_{a_i \in c_j} A_2(a_i)}{N_F^{S_2}}. \quad (12)$$

If $\max_{j=1}^k (\mu_{1j}) \geq \beta$ or $\max_{j=1}^k (\mu_{2j}) \geq \beta$ then the corresponding branch is terminated as a leaf, and this leaf is assigned as the class c_j . Otherwise, the S is partitioned into S_1 and S_2 :

$$\begin{aligned} S_1 &= \{s \mid A_1(a_i) \geq \lambda, a_i \in S\} \\ S_2 &= \{s \mid A_2(a_i) \geq \lambda, a_i \in S\} \end{aligned} \quad (13)$$

and the above steps are repeated for each child nodes until the above criterion (Eq.(11) or Eq.(12)) are satisfied. Usually, $\alpha = 0.1\sim 0.2$, $\beta = 0.8\sim 0.9$ and $\lambda = 0.5$ can be selected. Smaller α and bigger β will generate a bigger tree with higher accuracy in training. However, when the data is uncertain/noisy, too smaller α and too bigger β can cause the induced decision tree to become over-fitted.

5. Application to Machine Fault Diagnosis under Uncertainties

The proposed method has been successfully employed in an industrial application of diagnosing the conditions of tapping process. Tapping is a typical machining process to make threads inside a hole, which often suffers from operating problems. Usually, the malfunctions of the tapping process can be divided into five classes: (1) normal condition, (2) tap wear, (3) hole undersize, (4) hole oversize, and (5) misalignment. Diagnosis of the tapping process is required to ensure the quality of products [17,18]. Totally, 8 features were extracted: (1) the peak torque (x_1); (2) the mean torque (x_2); (3) the variance of the torque (x_3); (4) the mean of torque in retraction stroke (x_4); (5) the mean of thrust force (x_5); (6) the covariance of torque and thrust force (x_6); (7) the correlation of torque and thrust force (x_7) and (8) the correlation of torque and thrust force in retraction stroke (x_8). To show the effectiveness of the proposed method, a set of experiments under different machining conditions has been performed. A total of 80 training and 40 testing examples were collected independently.

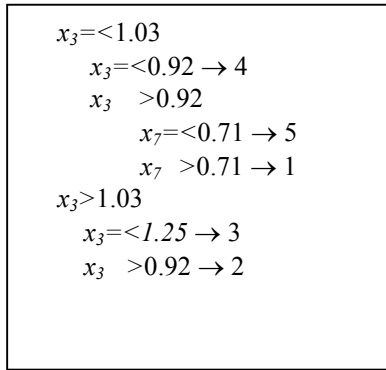


Fig. 3. Classical decision tree.

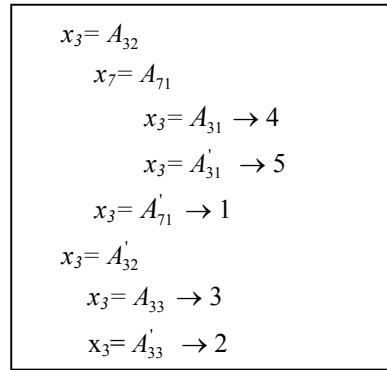


Fig. 4. Decision tree with soft discretization

The C4.5 decision tree learning algorithm has been applied to generate the classification rules from the given 80 training examples. The generated decision tree is shown in Fig.3. Using this decision tree, all the 80 training data points can be classified correctly, however, three out of 40 test examples are misclassified, as shown in Table.1. That means, even if a high accuracy has been obtained in training, the performance in testing can not be guaranteed yet.

Let α be 0.1, β be 0.8 and λ be 0.5, a soft discretization based decision tree has been induced using the presented method, as shown in Fig.4. The corresponding soft discretization, defined by fuzzy set pair, is shown in Fig. 5. Both the 80 training data and 40 testing data have been used to verify the classification capability of the generated decision tree. The possibilities ($\pi_j, j=1\sim 5$) of each data point belonging to each class (c_j) are calculated according to Eq.(1). The class having the maximum value of possibility (as underlined) is selected to represent the machining condition. The result for testing data is tabulated in Table 1. For comparison, the results obtained by C4.5 and C4.5 with boosting are shown in Table.1 also. It is showed that all the training and testing data have been correctly classified by the soft discretization based decision tree. Moreover, the results of soft-discretization decision tree are more close to human reasoning, and, more important, which is more reliable in decision-making.

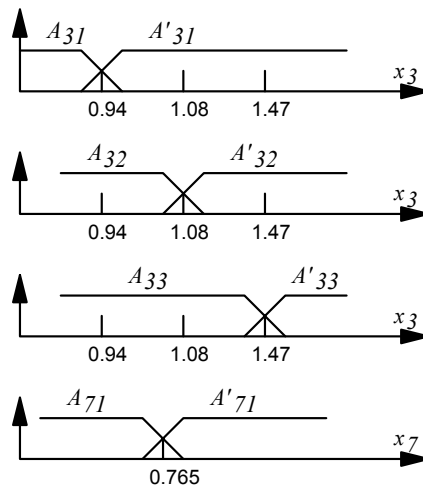


Fig. 5. Soft discretization of attributes

Table 1. Classification results of testing data

No	Class	π_1	π_2	π_3	π_4	π_5	Soft discretization	C4.5	C4.5 boosting
1	1	<u>1</u>	0	0	0	0	1	1	1
2	1	<u>0.9167</u>	0	0	0	0.0833	1	1	1
3	1	<u>1</u>	0	0	0	0	1	1	1
4	1	<u>1</u>	0	0	0	0	1	1	1
5	1	<u>1</u>	0	0	0	0	1	1	1
6	1	<u>1</u>	0	0	0	0	1	1	1
7	1	<u>1</u>	0	0	0	0	1	1	1
8	1	<u>1</u>	0	0	0	0	1	1	1
9	2	0	<u>1</u>	0	0	0	2	2	2
10	2	0	<u>1</u>	0	0	0	2	2	2
11	2	0	<u>1</u>	0	0	0	2	2	2
12	2	0	<u>1</u>	0	0	0	2	2	2
13	2	0	<u>1</u>	0	0	0	2	2	2
14	2	0	<u>1</u>	0	0	0	2	2	2
15	2	0	<u>1</u>	0	0	0	2	2	2
16	2	0	<u>1</u>	0	0	0	2	2	2
17	3	0	0	<u>1</u>	0	0	3	3	3
18	3	0	0	<u>1</u>	0	0	3	3	3
19	3	0	0	<u>1</u>	0	0	3	3	3
20	3	0	0	<u>1</u>	0	0	3	3	3
21	3	0	0	<u>1</u>	0	0	3	3	3
22	3	0	0	<u>1</u>	0	0	3	3	3
23	3	0	0	<u>1</u>	0	0	3	3	3
24	3	0	0	<u>1</u>	0	0	3	3	3
25	4	0	0	0	<u>1</u>	0	4	4	4
26	4	0	0	0	<u>1</u>	0	4	4	4
27	4	0	0	0	<u>1</u>	0	4	4	4
28	4	0	0	0	<u>1</u>	0	4	4	4
29	4	0	0	0	<u>1</u>	0	4	4	4
30	4	0	0	0	<u>1</u>	0	4	4	4
31	4	0	0	0	<u>1</u>	0	4	4	4
32	4	0	0	0	<u>1</u>	0	4	4	4
33	5	0	0	0	0	<u>1</u>	5	5	5
34	5	0	0	0	0	<u>1</u>	5	5	5
35	5	0	0	0	0	<u>1</u>	5	5	5
36	5	0.25	0	0	0	<u>0.75</u>	5	<u>1</u>	<u>1</u>
37	5	0	0	0	0	<u>1</u>	5	5	5
38	5	0	0	0	0	<u>1</u>	5	<u>3</u>	<u>3</u>
39	5	0	0	0	0	<u>1</u>	5	5	5
40	5	0.25	0	0	0	<u>0.75</u>	5	<u>1</u>	<u>1</u>

5. Conclusions

In the induction of continuous decision trees, the key point is to generate an appropriate attribute discretization at a node. The methods used in classical tree induction are noise sensitive, and thus prone to misclassification. Robustness and generalization capabilities are the major issues of continuous decision tree.

This paper concentrates on the fuzzy set based approach for the enhancement of decision trees. This paper presents an alternative method, called soft discretization, to generate a fuzzy decision tree. Based on defining soft discretization, using the concept of fuzzy partition and the criterion for finding the best soft discretization, a method of decision tree induction is presented in detail. The experimental results showed that higher accuracy of classification was obtained in testing by means of the presented method than of classical decision trees. The experiment results suggest that the use of soft discretization could have the potential capability to improve the robustness of classification and enhance the generalization of the induced classifier.

Though the effectiveness of the proposed approach has been illustrated in a particular industrial application, several theoretical issues are still open. Two of them are currently studied by the authors, that is, to design a method to automatically generate suitable membership parameters, i.e. to determine the width of the overlapping of fuzzy set pair, according to the characteristics of the attributes, and to further evaluate its performances in dealing with noise and imprecise data using cross-valid method based on more datasets.

References

1. J R Quinlan, Induction of Decision Trees, *Machine Learning*, 1 (1986) 81-106.
2. J R Quinlan, Decision Trees and Decision Making, *IEEE Transactions on System, Man and Cybernetic*, 20 (1990) 339-346.
3. L Breiman, J H Friedman, et al, *Classification and Regression Trees*, Belmont, CA: Wadsworth, 1984.
4. U M Fayyad, and B I Keki, On the handling of Continuous-Valued Attributes in Decision Tree Generation, *Machine Learning*, 8 (1992) 87-102.
5. U M Fayyad, and B I Keki, Multi-interval Discretization of Continuous Valued Attributes for Classification Learning, *IJCAI-93*, (1993) 1022-1027.
6. J R Quinlan, Improved use of Continuous Attributes in C4.5, *Journal of Artificial Intelligence Research*, 4 (1996) 77-90.
7. W Buntine. *Learning Classification Trees*, *Statistics and Computing*, 2:63-73, 1992.
8. G A, Ouzden and A W William, Induction of Rules Subject to a Quality Constraint: Probabilistic Inductive Learning, *IEEE Trans. On Knowledge and Data Engineering*, 5 (1993) 979-984.
9. X H Wang, J R Hong, On the Handling of Fuzziness for Continuous-valued Attributes in Decision Tree Generation, *Fuzzy Sets and Systems*, 99 (1998) 283-290.
10. C H Wang, J F Liu, T P Hong, and S S Tseng, A Fuzzy Inductive Learning Strategy for Modular Rules, *Fuzzy Sets and Systems*, 103 (1999) 91-105.
11. C Z Janikow, *Fuzzy Decision Trees: Issues and Methods*, *IEEE Transactions on System, man and Cybernetic-Part B: Cybernetics*, 28 (1998) 1-14.
12. G J Klir and T A Folger, *Fuzzy Sets, Uncertainty and Information*, Prentice Hall 1988.

13. Y H Peng, T J Chen, and P W Tse, Fuzzy Inductive Learning in Data Mining and its Application to Machine Fault Diagnosis, 2000'AMSMA international conference, pp.158-163, Guangzhou, China, (2000) 158-163
14. L A Zadeh, Fuzzy Sets as a bases for a Theory of Possibility, Fuzzy Sets and Systems, 1 (1978) 3-28.
15. L A Zadeh, Probability Measures of Fuzzy Events, Journal of Mathematical Analysis and Applications, 23 (1968) 421-427.
16. D Dumitrescu, Fuzzy Measures and Entropy of Fuzzy Partitions, Journal of Mathematical Analysis and Applications, 176 (1993) 359-373.
17. Y B Chen. A Fuzzy Decision System for Fault Classification under High Levels of Uncertainty, Transactions of ASME Journal of Dynamic Systems, Measurement and Control. 117 (1995) 108-115.
18. P W Tse, D D Wang etc. Harmony Theory Yields Robust Machine Fault-Diagnostic Systems Based On Learning Vector Quantization Classifiers. Engineering Applications of Artificial Intelligence, 9 (1996) 487-498.