ELSEVIER

Decision Support Systems 21 (1997) 61–73

Decision Support Systems

FILM: a fuzzy inductive learning method for automated knowledge acquisition

Bingchiang Jeng [1], Yung-Mo Jeng, Ting-Peng Liang *

Department of Information Management, National Sun Yat-sen University, Kaohsiung, 80424, Taiwan

Abstract

Inductive learning that creates decision trees from a set of existing cases is useful for automated knowledge acquisition. Most of the existing methods in literature are based on crisp concepts that are weak in handling marginal cases. In this paper, we present a fuzzy inductive learning method that integrates the fuzzy set theory into the regular inductive learning processes. The method converts a decision tree induced from regular method into a fuzzy decision tree in which hurdle values for splitting branches and classes associated with leaves are fuzzy. Results from empirical tests indicate that the new fuzzy approach outperforms the popular discriminant analysis and ID3 in predictive accuracy. © 1997 Elsevier Science B.V.

Keywords: Inductive learning; Machine learning; Expert systems; Fuzzy sets

1. Introduction

Inductive learning that generates decision trees or decision rules from existing cases is an important approach for automated acquisition of expert knowledge. Applications have been reported in many areas such as stock prediction [1], credit card application [2], graduate admission [3], inventory accounting method choice [4], loan evaluation [5,6], and medical diagnosis [7,8]. Their findings indicate that knowledge induced from these methods are equally or more accurate than statistical discriminant analysis or other competing models in predicting new cases.

A typical inductive learning process includes three steps. First, each attribute domain is partitioned into segments so that boundaries differentiating classes can be determined. This step determines the hurdle values for different classes. In credit card analysis, for instance, we may find that the salary of credit card holders can be partitioned into two segments at US$30000. That is, if the salary of an applicant is greater than or equal to US$30000, then its credit is good. If the salary is less than US$30000, then its credit classification is bad.

Following the segmentation of attributes, the discriminant power of each attribute is analyzed. In a classic method called ID3 (Iterative Dichotomizer 3) [9], the partition and discriminant power of an attribute are determined by a measurement called *entropy*. The attribute with a higher entropy value is considered having a

* Corresponding author. E-mail: liang@mis.nsysu.edu.tw

[1] E-mail: jeng@mis.nsysu.edu.tw

0167-9236/97/$17.00 © 1997 Elsevier Science B.V. All rights reserved.
PII S0167-9236(97)00019-5

higher discriminant power. Finally, a decision tree or if–then rules are constructed. Heuristics are often used to arrange the sequence of decision attributes. For example, if we find salary to be more powerful than personal assets for credit analysis, then salary will be evaluated in front of personal assets in the resulting decision tree.

Although existing methods are promising in enhancing the knowledge acquisition process, they also have a couple of known shortcomings. First, the hurdle values for attribute segmentation are crisp, which is inconsistent with human information processing. Using our previous example of credit analysis, an applicant making US$30000 annually is considered good, but another person making US$29999 will be considered bad by our rule. It is obvious that the difference is not so sharp in the real world. Second, the crisp nature of the hurdle values also affects the robustness of the induced decision trees. Because attribute segmentation is determined by the training cases, the resulting knowledge model based on crisp rules is more sensitive to the noises in the training data.

In this paper, we propose an approach called the fuzzy inductive learning method (FILM) that integrates the fuzzy set theory into the tree induction process to overcome these limitations. A major advantage of the fuzzy approach is that it allows the classification process to be more flexible and the resulting tree to be more accurate due to the reduced sensitivity to slight changes of hurdle points. Our empirical studies confirm this hypothesis by showing that FILM can improve existing methods including discriminant analysis and ID3.

The remainder of the paper is organized as follows. First, basic concept of the inductive learning process is introduced. This is followed by an introduction to the fuzzy set concepts. Then, the fuzzy inductive learning method is presented in detail. Finally, empirical results and conclusions are presented.

2. Inductive learning

Induction is a process by which a knowledge structure can be created from a set of data to explain or predict the behavior of the data set. Early work of inductive learning can be traced back to 1966 when Hunt et al. [10] developed a method for induction. This method was later implemented and expanded by Paterson and Niblett [11] to create ACLS (A Concept Learning System) and by Quinlan [9,12] to develop the popular ID3. The original ID3 algorithm uses hurdle values of different attributes to partition recursively a set of training data into mutually exclusive subsets until each subset contains cases of the same class or no attribute is available for further decomposition. Given a set of cases $C=\{(v_{i1}, \ldots, v_{in}; g_i)|v_{ij} \in V_j$ where V_j is the domain of an attribute F_j and g_i is the class of case $i\}$, the algorithm is described as follows:

```
1. Set the root node as $C =$ the whole training data,
2. Given $C$, do
   2.1. For each numerical attribute, do
          - Find a value $x_j$ to decompose the training set into two subsets,
          - Calculate the entropy of the decomposition,
          - Choose the decomposition whose entropy value is the largest,
   2.2. For each categorical attribute, decompose $C$ by its classes and calculate its entropy,
   2.3. Choose the attribute $F^*$ whose entropy value is the largest after decomposition to break $C$ into
        mutually exclusive subsets, $C_i$, where $i = 1 \ldots k$,
   2.4. Label $F^*$ as the root node and subsets $C_i$ to its leaves,
3. For $i = 1$ to $k$, if cases in $C_i$ is not of the same class, then let $C = C_i$ and go to 2, otherwise stop.
```

Fig. 1 shows an example of the decision tree for bankruptcy analysis induced from a set of financial data. The tree indicates that decomposing the training cases at $F3$ (net income/total asset) = 0.0295 would give us the highest discriminant power. For cases whose $F3$ value is greater than 0.0295, decomposing by $F4$ (current assets/total assets) at 0.7487 would allow us to differentiate the bankrupt firms from safe ones.

The above exhaustive decomposition method has some known shortcomings. For instance, it often overfits the training data and, hence, becomes very sensitive to the noise in the training data. Data error directly affects the hurdle value u_j and therefore reduces the predictive accuracy of the resulting decision tree. In addition, the

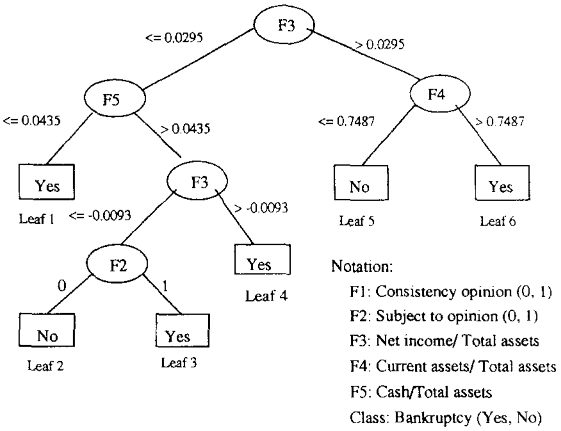


Fig. 1. A sample decision tree for bankruptcy prediction.

resulting tree tends to contain more nodes than necessary in order to cover a small number of unusual cases. The leaf nodes may also include inconsistent cases such as a few bankrupt cases in a node classified as non-bankrupt. Finally, for attributes whose domains are real numbers, it is computationally inefficient to find the optimal hurdle value that maximizes the entropy value due to theoretically unlimited number of possible decompositions.

Recently, several approaches have been developed to overcome the above problems. Quinlan [13] proposed a tree pruning method that evaluated the contribution of each node after tree construction and removed unnecessary nodes to reduce the size of the tree and to increase its resistance to data noises. Quinlan [14] discussed the induction of probabilistic decision trees. Cleary [15] presented an approach that incorporated probabilities when trees or decision rules were constructed. This allows inconsistent cases in a node to be handled by rule uncertainty. Liang [16] developed a composite approach to rule induction that processed nominal and non-nominal attributes differently. Probabilistic information is used to analyze non-nominal attributes to accommodate data noises and to obtain more accurate hurdle values. Mookerjee and Dos Santos [17] proposed an approach that determines the optimum decision tree based on information costs. Liang, et al. [24] suggested an approach that integrates statistical and inductive learning methods.

Although the above works have solved some of the problems, there is another issue that has not been addressed adequately by the existing work. That is, whether the hurdle values that decompose the training data into subsets should be crisp. In the real world, many decision boundaries are fuzzy and marginal differences are often ignored. A crisp decision tree is easy to read and to follow. However, marginal cases are often misclassified due to the non-compensatory nature of the decision tree (i.e., the weakness in one attribute cannot be offset by the strength in another). For example, case $A = (0, 1, 0.0296, 0.7487, 0.01)$ is classified into leaf 5 (non-bankrupt) of the tree in Fig. 1, whereas case $B = (0, 1, 0.0296, 0.7488, 0.01)$ is classified into leaf 6 (bankrupt). This classification is obviously vulnerable. Should the $F3$ value of A changes from 0.0296 to 0.0295, it would have been classified into leaf 1, which is bankrupt. This example indicates that the crisp hurdle value employed in traditional inductive learning methods may cause problems and must be handled in different ways [18]. In the following sections, we present a fuzzy approach to alleviate the above problem.

3. Fuzzy sets theory

The fuzzy set theory was developed by Zadeh in 1965 [19] and later extended and applied to many fields, including artificial intelligence. For instance, a recent work by Jeng and Liang [20] applied fuzzy sets to the retrieval and indexing of cases in case-based systems. The primary purpose of the fuzzy set theory is to provide a method by which qualitative terms with ambiguous meanings such as *old*, *tall*, and *very beautiful* can be modelled.

The key concept of fuzzy sets is to give a membership degree to each set member. In a classical set (usually called a *crisp* set), whether an object belongs to the set or not is binary. That is, the object either belongs to the set completely or does not belong to it at all. There is nothing in between. In the real world, however, a lot of things are ambiguous. For instance, a lady may be a beauty in some sense but not completely. The degree that an object belongs to a set is called its *membership degree*. The function that generalizes the membership degrees of all members in a set is called its membership function. The range of a membership function is the interval between 0 and 1. In other words, the maximum membership degree an object may have is 1 and the minimum is 0. A membership function μ_s can be represented below:

$$\mu_s : X \to [0,1].$$

Given the membership function, a fuzzy set S is represented as $\{(x, \mu_s(x)) | x \in U$, where U is the domain of $x\}$. The membership function of a fuzzy set is often a continuous function of its attribute values. For example, the membership function of *old* may be a function of age as shown in Fig. 2. In the figure, we can see that any person older than 50 can be considered old to some degrees. A 60-year-old person may be considered old with a membership degree of 0.7, whereas a 65-year-old person is old with a membership degree of 1.0.

The major advantage of using fuzzy sets is that memberships can be represented in a more flexible way. It allows information unavailable in crisp sets to be included. In fact, crisp sets are special cases of fuzzy sets. A fuzzy set can easily be converted into a crisp set. One popular approach is to use a hurdle value α, called α-cut, to differentiate memberships. Fuzzy members whose membership degrees are greater than or equal to α remain in the set and all others lose their memberships. The converted crisp set S_α is:

$$S_\alpha = \{x \epsilon U | \mu_s(x) \geq \alpha\}.$$

Similar to crisp sets, fuzzy sets can be manipulated by many operators such as equality, inclusion, projection, join, union, and intersection. Two operations of particular interests to inductive learning are set *union* and *intersection*. The union of two sets is a superset in which the membership degree of a member is the maximum of its membership degrees in individual sets. Formally, we define as follows:

$$A \cup B \Leftrightarrow \forall x \in U, \mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x)). \qquad (1)$$

The intersection of two fuzzy sets results in a new set whose membership degrees are the minimum of their individual degrees in the two sets. Formally, we define as follows:

$$A \cap B \Leftrightarrow \forall x \in U, \mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x)). \qquad (2)$$

Sometimes, the above definitions have problems in capturing all relevant information. For instance, the

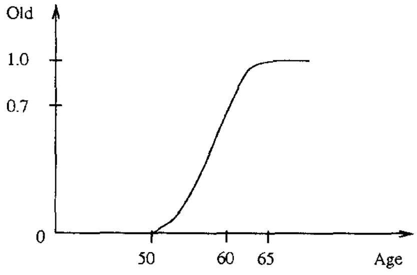


Fig. 2. Membership function of OLD.

resulting membership degree of $A \cap B$ is the same (both are 0.4) for $[\mu_A(x) = 0.8, \mu_B(x) = 0.4]$ and $[\mu_A(y) = 0.4, \mu_B(y) = 0.4]$. Obviously, the information that $\mu_A(x)$ is greater than $\mu_B(x)$ is lost.

Another approach to overcome the problem is to use a family of functions known as *Yager class* to modify the operations [21]. Yager class uses a parameter w to control the strength of the resulting membership function, where $1 < w < \infty$. The union and intersection functions can be redefined as follows:

$$\mu_{A \cup B}(x) = \min\left(1, \left(\mu_A(x)^w + \mu_B(x)^w\right)^{1/w}\right) \tag{3}$$

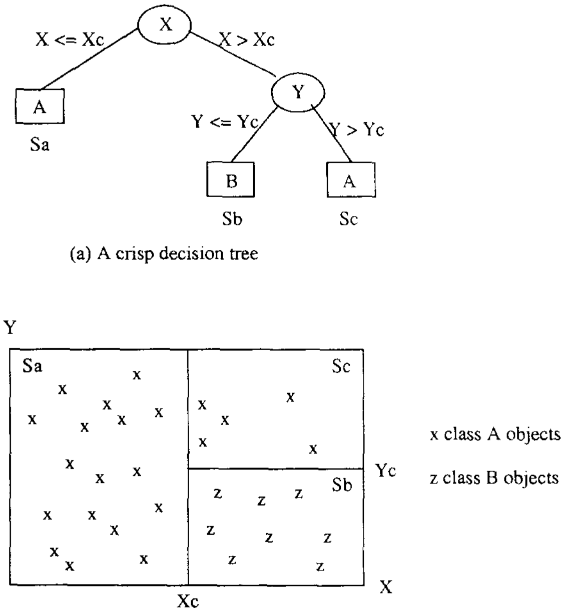
$$\mu_{A \cap B}(x) = 1 - \min\left(1, (1-\mu_A(x))^w + \left(1-\mu_B(x)^w\right)^{1/w}\right). \tag{4}$$

These two functions are general forms of our previous definitions. When $w = \infty$, the Yager functions of $\mu_{A \cup B}(x)$ and $\mu_{A \cap B}(x)$ become our original definitions, that is, $\max(\mu_A(x), \mu_B(x))$ and $\min(\mu_A(x), \mu_B(x))$ [21]. Since Yager functions provide more flexibility, we employ them in our tree induction method.

4. Fuzzy tree induction

In some sense, the induction of decision trees can be seen as a process by which the attribute space is partitioned to maximize the internal similarity within subspaces. For example, a decision tree as shown in Fig. 3a partitions the space of two attributes, (X, Y), into three subspaces as shown in Fig. 3b. Subspaces S_a and S_c contain cases of class A. Subspace S_b contains cases of class B. In a traditional decision tree, the hurdle values of x_c and y_c are crisp. In other words, any case whose x value is smaller than x_c is classified into class A no matter how small is the difference.

The fuzzy tree induction process uses the crisp decision tree generated from ID3 or other methods as a base and then applies a fuzzification operation to modify it. The fuzzification operation includes two steps. First, it begins with the fuzzification of the hurdle value. Instead of creating single values for partitioning the space, membership functions are also assessed for each attribute to give fuzzy subspace borders. Once the hurdle



(a) A crisp decision tree

(b) Partition of attribute space

Fig. 3. Space partition by a decision tree.

values are fuzzified, we can determine the possibility that a case belongs to a leaf node and reassign the class of each subspace. The original crisp decision tree becomes fuzzy at this stage. Therefore, a complete fuzzy decision tree is composed of fuzzy hurdle values and fuzzy classes. When it is applied to analyze a new case, a defuzzification process must be used to convert fuzzy classes into a conclusion. In the following sections, we shall describe the fuzzification and defuzzification processes in detail.

4.1. Fuzzification of a hurdle value

The fuzzification of a hurdle value changes the value from *exactly* x_{c} to *about* x_{c}. We use $\underline{x}_{\mathrm{c}}$ to stand for the fuzzy value of x_{c}. Accordingly, the relationships of $x \leq x_{\mathrm{c}}$ and $x > x_{\mathrm{c}}$ can also be changed to $x \leq \underline{x}_{\mathrm{c}}$ and $x > \underline{x}_{\mathrm{c}}$, respectively.

The fuzzy membership functions of $x \leq \underline{x}_{\mathrm{c}}$ and $x > \underline{x}_{\mathrm{c}}$ can be defined in many different ways. A straightforward approach is to assume a linear function between the upper and lower bounds of x. Their definitions are as follows:

$$\mu_{x \leq \underline{x}_{\mathrm{c}}}(x) = \begin{cases} 1, & \text{if } x \leq x_{\mathrm{cl}} \\ (x_{\mathrm{cu}} - x)/(x_{\mathrm{cu}} - x_{\mathrm{cl}}), & \text{if } x_{\mathrm{cl}} \leq x \leq x_{\mathrm{cu}} \\ 0, & \text{if } x_{\mathrm{cu}} \leq x. \end{cases} \tag{5}$$

$$\mu_{x > \underline{x}_{\mathrm{c}}}(x) = \begin{cases} 0, & \text{if } x \leq x_{\mathrm{cl}} \\ (x - x_{\mathrm{cl}})/(x_{\mathrm{cu}} - x_{\mathrm{cl}}), & \text{if } x_{\mathrm{cl}} \leq x \leq x_{\mathrm{cu}} \\ 1, & \text{if } x_{\mathrm{cu}} \leq x. \end{cases} \tag{6}$$

Fig. 4 shows the graphical form of the membership functions. Suppose $x_{\mathrm{cl}} = 0$ and $x_{\mathrm{cu}} = 10$, then $\mu_{x \leq \underline{x}_{\mathrm{c}}}(x = 8) = 0.2$ and $\mu_{x > \underline{x}_{\mathrm{c}}}(x = 8) = 0.8$.

4.2. Fuzzy classification of tree leaves

Each leaf of a decision tree is a subspace as shown in Fig. 3, which is formed by a set of conjunctive conditions. After fuzzifying the conditions associated with attributes, we can define the fuzzy subspace. The crisp borders as shown by solid lines are replaced by fuzzy borders as shown by gray areas in Fig. 5. In the figure, fuzzy borders are the result of fuzzy operations applied to fuzzy hurdle values of the attributes. For example, the possibility that a case (x, y) is in subspace S_{b} is the conjunction of the possibilities of $x > \underline{x}_{\mathrm{c}}$ and $y \geq \underline{y}_{\mathrm{c}}$. That is,

$$\mu_{S_{\mathrm{b}}}(x, y) = \mu_{x > \underline{x}_{\mathrm{c}}}(x) \cap \mu_{y \leq \underline{y}_{\mathrm{c}}}(y) \tag{7}$$

If two attributes X and Y are both fuzzified, then we can apply Eq. (2) or Eq. (4) to calculate the

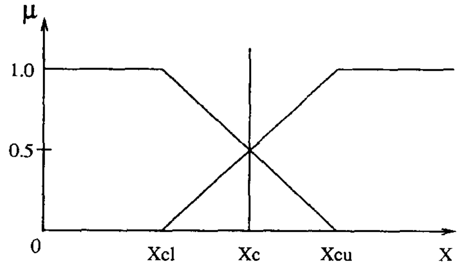


Fig. 4. A fuzzy boundary.

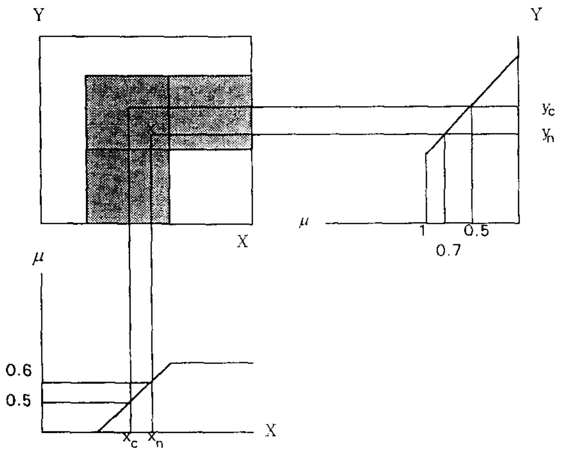


Fig. 5. Fuzzy association of an object.

membership degree that an observation falls into a subspace. If we use Eq. (2) for simplicity, then a case having $\mu_{x > \underset{\sim}{x_c}}(x) = 0.6$ and $\mu_{y \leq \underset{\sim}{y_c}}(y) = 0.8$ will have a possibility of 0.6 (the minimum of 0.6 and 0.8) to be classified into subspace S_b. Here, the possibility is its membership degree.

4.3. Construction of fuzzy decision trees

The final step of fuzzy tree induction is to fuzzify the crisp decision tree so that the hurdle values and leaf classifications are both fuzzy. For example, the crisp decision tree shown in Fig. 3a must be converted to the fuzzy decision tree as shown in Fig. 6. This stage consists of two steps: reclassification of training cases and calculation of class memberships.

4.3.1. Reclassification of training cases

Since the crisp hurdle values have been replaced by fuzzy values, all training cases must be analyzed using the procedures described in Sections 4.1 and 4.2 to reassess their leaf associations. Please note that a case now may associate with more than one leaf (with different membership degrees) of the decision tree. For example, a case may be assessed to have 0.6 possibility belonging to leaf S_a, 0.3 possibility belonging to leaf S_b, and 0.8 possibility belonging to leaf S_c. The major operation for this step is set intersection. Either the simple equation in Eq. (2) or Yager function in Eq. (4) can be used.

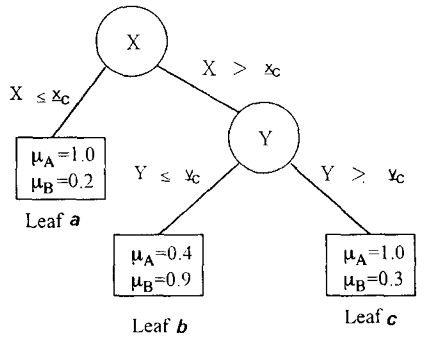


Fig. 6. A fuzzy decision tree.

4.3.2. Calculation of class memberships

After obtaining all possible leaf associations of the training cases, we further calculate the class association of leaves (that is, whether a particular leaf in Fig. 6 belongs to class A or B). Again, the class association of a leaf may not be unique. It may associate with different classes with different membership degrees. The calculation procedure is simple. We apply the union operation that combines the possibilities of all cases of the same class in a leaf to obtain the class association of the leaf. For example, each case has a known possibility to associate with a leaf after step (1). For leaf S_i, the possibility that the leaf associates with class A or B (i.e., μ_A or μ_B) is the union of all possibilities that the cases of class A or B falling in the leaf.

Formally, we can define as follows (C_a and C_b stand for all cases of classes A and B respectively):

$$\mu_A(S_i) = \cup_{C_a} \mu_{S_i}(x,y), \text{ for all cases } (x,y) \text{ in } C_a \tag{8}$$

$$\mu_B(S_i) = \cup_{C_b} \mu_{S_i}(x,y), \text{ for all cases } (x,y) \text{ in } C_b \tag{9}$$

Fig. 7 shows a fuzzy decision tree induced from the bankruptcy data listed in Appendix A. Each leaf shows the ID3 classification and FILM classification. For instance, leaf 1 indicates that the ID3 classification of those cases whose $F4 \leq 0.02955$ and $F8 \leq 0.04255$ is *yes* (i.e., bankrupt). The FILM classification is 1.0 possibility to be *yes* and 0.394 possibility to be *no*. Detailed calculation in the example was performed by a prototype system implemented in Turbo C and is too complicated to show here.

4.4. Prediction of new cases

Since most information in a fuzzy decision tree is fuzzy, applying it to predict the class of a new case is a little more complicated than using a crisp tree. In general, the prediction process includes two steps: feature mapping and defuzzification.

Feature mapping requires that the new case be mapped to the fuzzy attribute space. The attribute values are used to obtain the membership degree that the case belongs to a particular leaf. The mapping is one to many. That means, a case can have more than one mapped leaf.

The procedure for mapping new cases is the same as those presented in Section 4.3 regarding the reclassification of training cases. They will have different degrees of association with each leaf. For example, the associations of case #3 in Appendix A with leaves in Fig. 7 are 0.521 for leaf 4, 0.479 for leaf 5 and 0 for the rest.

After obtaining the leaf association, we calculate the class association of the case (i.e., the possibility that the case belongs to a certain class) at each leaf. This is done by multiplying the leaf association with the class association. The results for leaves 4 (L4) and 5 (L5) are as follows (the rest nodes are zero that can be ignored):

L4: $\mu_{\text{no}}(C3) = 0.521 \times 0.376 = 0.196$; $\mu_{\text{yes}}(C3) = 0.521 \times 0.899 = 0.468$.

L5: $\mu_{\text{no}}(C3) = 0.479 \times 1.000 = 0.479$; $\mu_{\text{yes}}(C3) = 0.479 \times 0.344 = 0.165$.

Finally, we need a defuzzification mechanism to conclude exactly which class the case belongs. The results obtained from the previous stage are often contradictory. For example, the above data indicates that L4 suggests the case be classified as bankrupt, whereas L5 suggests the opposite.

Again, there are many different ways for defuzzification. The simplest one we use is called the K-sum approach. The approach consists of two steps. First, we choose the highest K values of class association. Then, the chosen values are summed up by classes. The class with the highest sum is concluded to be the class of the new case. Here, K is an integer. In our previous example, different Ks give us the same conclusion, as shown below:

$K = 1$: No; ($\mu_{\text{no}} = 0.479$)

$K = 2$: No; ($\mu_{\text{no}} = 0.479 > \mu_{\text{yes}} = 0.468$)

$K = 3$: No; ($\mu_{\text{no}} = 0.479 + 0.196 = 0.675 > 0.468$)

$K = 4$: No; ($\mu_{\text{no}} = 0.479 + 0.196 = 0.675 > 0.468 + 0.165 = 0.633$)

Therefore, the FILM's classification of case #3 should be *no*, which is correct. If you use the original ID3 decision tree, you will find that the case falls into leaf 4. This gives us an incorrect classification of *yes*.

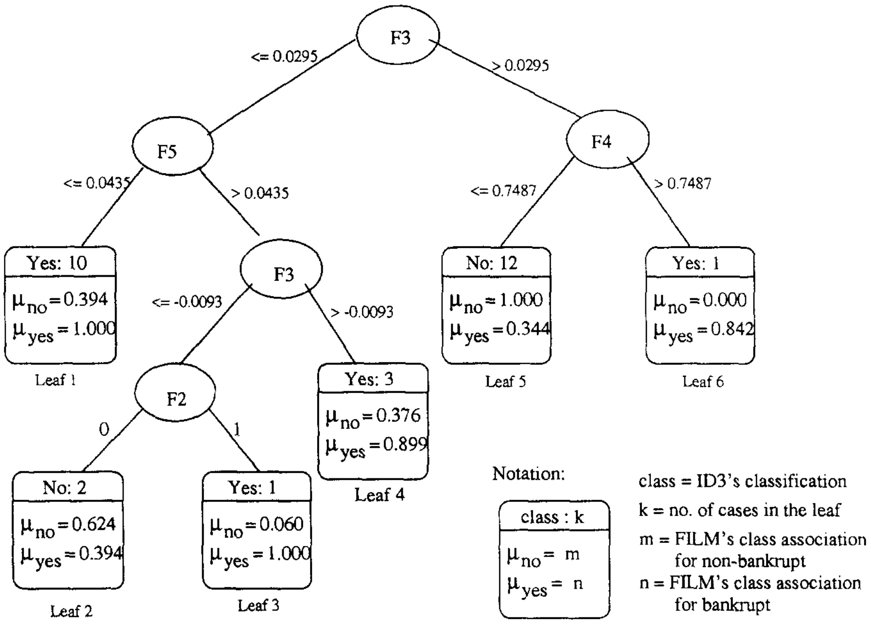


Fig. 7. A large fuzzy decision tree.

Other criteria that may be used for defuzzification include total value sum (summarize possibilities of all leaves and choose the class with the highest value), value sum of k-nearest neighbors (summarize the possibilities of k nearest neighbors and choose the class with the highest total possibility value), majority class of k-nearest neighbors (choose the majority class among the highest k possibilities), sum value above α-cut (choose an α-cut and the class whose possibility sum is greater than the α-cut) and majority class above α-cut (choose the majority class among the leaves above a predefined α-cut), and so forth.

5. Empirical evaluation

In Section 4, we have presented the process of FILM. To further understand the performance of FILM, we apply it to analyze eight sets of data.

5.1. Data sets

The data sets employed for evaluation are obtained from various sources. Their features are briefly described below and summarized in Table 1.

(1) Bankruptcy data: This set was used by Liang [16] to evaluate the CRIS method. The whole set contains 30 cases. Each case is composed of eight attributes, three of which are categorical. The dependent attribute is either bankrupt or not.

(2) Iris data: This was the original data set Fisher used to illustrate the discriminant analysis [22]. The set contains 150 cases of three different kinds of flowers. Each case consists of four numerical attributes.

(3) Biomedical data: This set uses four different blood tests to differentiate infected from normal persons. The original set contained 209 cases (134 normal and 75 infected). After removing those with missing values, 194 were actually used in our experiment.

(4) Breast cancer data: This data set consists of 286 cases about predicting whether the patient will be ill again after treatment. Each case is represented by nine attributes. This was originally used by Michalski's AQ15 method and later used for comparing different methods [23].

Table 1
Summary of training data sets

Data sets	Domain	Classes	Categorical attributes	Quantitative attributes	Training samples
Iris	Biological	3	0	4	150
Appendicitis	Medical	2	0	7(8)[a]	106
Breast cancer	Medical	2	5	4	286
Wisconsin breast cancer	Medical	2	0	9	683(699)[b]
Pima Indians diabetes	Medical	2	0	8	768
Blood	Medical	2	0	5	194(205)[b]
Bankruptcy	Financial	2	3	5	30
Simulated data	None	2	0	4	200

[a] Only seven out of eight quantitative attributes were used in the experiments due to missing values.

[b] Numbers in front of parentheses indicate the actual number of training cases used in the experiments.

(5) Wisconsin breast cancer data: This data set contains 699 cases regarding the diagnosis of breast cancers collected from the University of Wisconsin Hospitals by Dr. W.H. Wolberg. Each case consists of nine integer attributes. The actual number of cases used was 683 because cases containing missing values were removed in our experiment.

(6) Appendicitis data: The data set contains 106 cases related to the diagnosis of appendicitis (85 positive cases). The original case is composed of eight attributes [7]. In our experiment, we removed one that had many missing values to use only seven attributes.

(7) Simulation data: This set contains 200 cases generated by the computer. Each case is composed of two categorical and four real number attributes.

(8) Pima Indians diabetes data: This set contains 768 cases related to the diagnosis of diabetes (268 positive and 500 negative). Each case is composed of eight numerical attributes.

5.2. Experimental procedures

The treatment of the experiment is different induction methods. Three methods were compared: statistical discriminant analysis (DA), ID3 and FILM. We use DA and ID3 as the benchmark for evaluating FILM.

We use predictive accuracy to compare the performance of these methods. Except the breast cancer, Wisconsin breast cancer, and Pima data sets, the predictive accuracy was measured using the leaving-one-out approach. That is, for a data set containing n cases, n-1 of them will be used as training data. The induced tree of discriminant function is then applied to predict the remaining one case. The above procedures are repeated n times. The overall predictive accuracy is the average of the n times. The result obtained from this method is generally considered to be very close to the actual predictive accuracy [23].

For the breast cancer data, we randomly chose 70% of the data as the training set and held the remaining 30% for cross-evaluation. This procedure was repeated four times. The predictive accuracy is the average of the four. The Wisconsin breast cancer and Pima data sets were evaluated using 10-fold cross-evaluation. One-tenth of the data set is held for testing and the remaining nine-tenths are used for training. The procedure repeats 10 times. These two methods were chosen primarily because these data sets have a large number of cases, which prevent them from using leaving-one-out method efficiently.

We control two parameters of FILM during the experiment: Yager's w and the width of the fuzzy border (i.e., $x_{\text{cu}} - x_{\text{cl}}$). We experimented with different w values ranging from 3 to 7 to obtain better results. The width of the fuzzy border was determined by the standard deviation of the training sample. We experimented with widths between 0.4 to 0.9 standard deviations.

5.3. Results and discussions

Table 2 shows the predictive accuracy of each method when applied to different data sets. The Yager's w was the w value we actually used in the experiment to achieve the accuracy. The average predictive accuracies

Table 2
Prediction accuracy of different methods

Data sets	DA	ID3	FILM	Yager's w
Iris	0.980	0.940	0.980	3
Appendicitis	0.858	0.802	0.887	3/4/5/6/7
Breast cancer	0.744	0.672	0.736	3/4
W. breast cancer	0.959	0.946	0.971	3/4/6
P.I. diabetes	0.764	0.720	0.776	6/7
Blood	0.866	0.809	0.907	4
Bankruptcy	0.700	0.767	0.833	6/7
Simulated	0.690	0.715	0.765	3

are 0.820 for DA, 0.796 for ID3, and 0.857 for FILM. The result is obvious: FILM outperformed both DA and ID3. Comparing FILM with DA, FILM outperformed DA in six data sets and was equal to DA in one. DA outperformed FILM only in one data set. Comparing FILM with ID3, FILM outperformed ID3 in all eight experiments. A paired t-test indicates that the difference is statistically significant at 5% level ($t = 2.58$). Therefore, we can conclude that *FILM is significantly better than ID3.*

FILM is a post-treatment of traditional tree induction methods. In this section, we have seen that the original ID3 method can be significantly improved in its predictive accuracy after processing by FILM. However, FILM is not without problems. During the experiment, a major problem we found was how to determine the optimum Yager's w and how to choose a proper membership function for an attribute. In the experiment, we used the trial-and-error approach to find the optimum w and fuzzy range. This is inefficient and can be further improved. Based on the results, it seems that Yager's w is usually good at 3 or 4, but 6 or 7 is also useful in some cases.

The reason that the fuzzy tree generated by FILM was more accurate than the original tree of ID3 is that the former processes marginal cases more accurately. A crisp decision tree is non-compensatory in predicting new cases. That is, the strengths in one attribute cannot be used to compensate the slight weakness in another. The fuzzy set concept allows the decision tree to be compensatory to some extent. This increases its flexibility in handling border values.

6. Conclusions

Tree induction has been a major technique for automated knowledge acquisition. Most existing tree induction methods are crisp in nature. They create decision trees with crisp hurdle values and deterministic class association of each leaf. In this paper, we have presented a new approach that applies the fuzzy set concepts to enhance the predictive accuracy of the induced tree. We first described the need for such an approach. Then, we illustrated the mechanism including fuzzification of hurdle values, tree leaves, and class associations. Finally, we presented experimental results that showed FILM outperformed the popular ID3 approach significantly.

Although our results show the potential of FILM, there are some issues that require further research. First, whether fuzzy treatment is good for ID3 only or it can be applied to other methods to enhance predictive accuracy. We plan to conduct experiments that test FILM's generalizability to other kind of trees. Second, the determination of Yager's w and the upper and lower bounds for fuzzy conversion of attributes is still largely heuristic in nature. In the future, we may be able to find a better way to determine these key parameters. Finally, a good tree induction method should be able to tolerate a certain degree of data noise in the training data set. We need to study how data noises affect the accuracy of the tree generated by FILM and other methods.

Appendix A. Bankrupt data set

ID	Class	*F*1	*F*2	*F*3	*F*4	*F*5	*F*6	*F*7	*F*8
1	No	0	0	0	0.0490	0.5327	3.5828	0.7577	0.0371
2	No	0	0	0	−0.0148	0.7128	1.9285	0.8425	0.0673
3	No	0	0	0	0.0286	0.2699	1.6437	0.6583	0.1753
4	No	0	0	0	0.1007	0.5773	6.5540	0.6837	0.0675
5	No	0	1	0	0.0365	0.2170	1.9699	0.8573	0.0169
6	No	0	0	0	0.0496	0.1497	1.5423	0.2961	0.0590
7	No	0	0	0	0.0411	0.6808	4.1449	0.6622	0.0234
8	No	0	0	0	0.0800	0.5203	5.8925	0.7986	0.0862
9	No	0	0	0	0.0994	0.2042	1.1392	0.4535	0.0577
10	No	0	0	0	0.0676	0.5483	2.3371	0.6255	0.0196
11	No	0	0	0	0.2099	0.4380	4.3876	0.7188	0.1893
12	No	0	1	0	−0.2306	0.4261	1.8098	0.4207	0.0455
13	No	0	0	0	0.1066	0.3944	2.9197	0.6750	0.0434
14	No	0	0	0	0.1295	0.3930	3.4351	0.6493	0.1209
15	No	0	0	0	0.0509	0.6313	3.7568	0.06675	0.0438
16	Yes	0	1	0	0.0226	0.3130	0.9602	0.6729	0.0136
17	Yes	0	0	1	−0.1005	0.1202	0.1709	0.1105	0.0041
18	Yes	0	0	0	0.0050	0.1290	1.0558	0.4428	0.0208
19	Yes	0	1	1	−0.2746	0.2787	1.8978	0.9225	0.0547
20	Yes	0	0	0	−0.1324	0.5500	1.8763	0.4340	0.0402
21	Yes	0	1	0	−0.0645	1.5557	2.9152	0.1961	0.0145
22	Yes	0	0	0	0.0189	0.2409	1.2443	0.5667	0.0303
23	Yes	1	1	1	−0.1953	0.0113	0.0015	0.0013	0.0013
24	Yes	0	0	0	−0.1356	0.4794	2.4443	0.5497	0.0416
25	Yes	0	1	0	−0.0038	0.6956	1.9334	0.8562	0.0937
26	Yes	0	0	1	0.0118	0.9479	0.1530	0.0902	0.0902
27	Yes	0	0	0	0.0029	0.3398	1.8195	0.9014	0.1704
28	Yes	0	1	0	0.0448	0.8165	1.4482	0.7506	0.0286
29	Yes	0	0	1	−0.1046	0.7100	1.1111	0.7660	0.0233
30	Yes	0	0	1	−0.0569	0.3652	2.2768	0.6655	0.0069

References

[1] H. Braun, J.S. Chandler, Predicting stock market behavior through rule induction: an application of the learning-from-example approach, Decision Sci. 18 (3) (1987) 415–429.

[2] C. Carter, J. Catlett, Assessing Credit Card Applications Using Machine Learning, IEEE Expert (1987) 71–79.

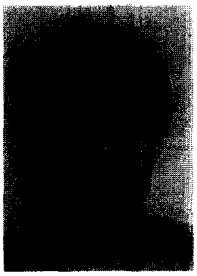
[3] H.M. Chung, M.S. Silver, Rule-based expert systems and linear models: an empirical comparison of learning-by-examples methods, Decision Sci. 23 (1992) 687–707.

[4] T.P. Liang, J.S. Chandler, I. Han, R. Roan, An Empirical investigation of some data effects on the classification accuracy of probit, ID3 and neural networks, Contemp. Accounting Res. 9 (1) (1992) 306–328.

[5] W.F. Messier Jr., J.V. Hansen, Inducing rules for expert system development: an example using default and bankruptcy data, Manage. Sci. 34 (12) (1988) 1403–1415.

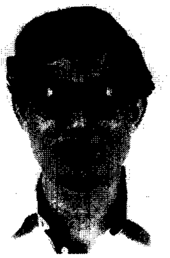
[6] M.J. Shaw, J.A. Gentry, Using an expert system with inductive learning to evaluate business loans, Financial Manage. 17 (3) (1988) 45–56.

[7] A. Marchand, F. VanLente, R. Galen, The assessment of laboratory tests in the diagnosis of Acute appendicitis, Am. J. Clin. Pathol. 80 (3) (1983) 369–374.
[8] R. Michalski, I. Mozetic, J. Hong, N. Lavrac, The Multi-purpose Incremental Learning System AQ15 and Its Testing Application to Three Medical Domains, Proceedings of the 5th Annual National Conference on Artificial Intelligence (1986) 1041–1045.
[9] J.R. Quinlan, Discovering rules from large collections of examples: A Case Study, in: D. Michie (Ed.), Expert Systems in the Micro Electronic Age, Edinburgh University Press, Edinburgh, Scotland, 1979.
[10] E.B. Hunt, J. Marin, P.T. Stone, Experiments in Induction, Academic Press, New York, NY, 1966.
[11] A. Paterson, T. Niblett, ACLS User Manual, Intelligence Terminal, Glasgow, Scotland, 1982.
[12] J.R. Quinlan, Induction of decision trees, Machine Learn. 1 (1) (1986) 81–106.
[13] J.R. Quinlan, Simplifying decision trees, Int. J. Man–Machine Studies 27 (1987) 221–234.
[14] J.R. Quinlan, Probabilistic decision trees, in: P. Langley (Ed.), Proceedings of the Fourth International Workshop on Machine Learning, Morgan Kaufman, Los Altos, CA, 1987.
[15] J.G. Clearly, Acquisition of uncertain rules in a probabilistic logic, Int. J. Man–Machine Studies 27 (1987) 145–154.
[16] T.P. Liang, A composite approach to inducing knowledge for expert system design, Manage. Sci. 38 (1) (1992) 1–17.
[17] V.S. Mookerjee, B.L. Dos Santos, Inductive expert systems design: maximizing system values, Inf. Systems Res. 4 (2) (1993) 111–140.
[18] B. Jeng, Y.M. Jeng, A Fuzzy Tree Induction Learning Method, The First Asian Fuzzy Systems Symposium, Nov., 1993.
[19] L.A. Zadeh, Fuzzy sets, Inf. Control 8 (1965) 338–353.
[20] B. Jeng, T.P. Liang, Fuzzy indexing and retrieval in case-based systems, Expert Systems with Applications 8 (1) (1995) 135–142.
[21] G.J. Klir, T.A. Floger, Fuzzy Sets, Uncertainty, and Information, Prentice-Hall, Englewood Cliffs, NJ, 1988.
[22] R.A. Fisher, The use of multiple measurements in taxonomic problems, Ann. Eugenics 7 (1936) 179–188.
[23] S.M. Weiss, C.A. Kulikowski, Computer Systems That Learn, Morgan Kaufman, San Mateo, CA, 1991.
[24] T.P. Liang, J.S. Chandler, I. Han, Integrating statistical and inductive learning methods for knowledge acquisition, Expert Systems with Applications 1 (4) (1990) 391–401.



Bingchiang Jeng received B.A. and M.S. degrees in Computer Engineering from National Chiao-Tung University in 1979 and 1981, respectively, and Ph.D. degree in Computer Science from New York University in 1990. He is currently an Associate Professor of the Department of Information Management, National Sun Yat-Sen University. His research interests are software testing, machine learning and software engineering.

Yung-Mo Jeng received an MBA degree in information systems from the Department of Information Management, National Sun Yat-Sen University, Kaohsiung, Taiwan. Machine learning has been his major research interest.



Ting-Peng Liang is a Professor in information systems and Dean of the College of Management, National Sun Yat-Sen University, Kaohsiung, Taiwan. Prior to joining the University in 1992, he had been on the faculty of the University of Illinois at Urbana-Champaign and Purdue University. He received his Ph.D. in information systems from The Wharton School, University of Pennsylvania. His research has been published in journals such as Management Science, Operations Research, Decision Sciences Support Systems, and IEEE Computers. He is also serving on the editorial board of more than 10 professional journals.