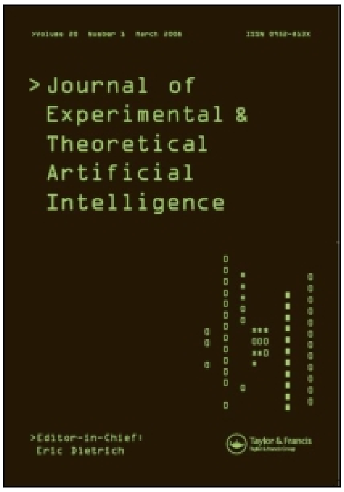

Journal of Experimental & Theoretical Artificial Intelligence
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t713723652

MFILM: a multi-dimensional fuzzy inductive learning method

Yao-Tsung Chen [a]; Bingchiang Jeng [b]
[a] Institute of Information Science, Academia Sinica, Taiwan [b] Department of Information Management, National Sun Yat-sen University, Taiwan

Online Publication Date: 01 September 2005

To cite this Article Chen, Yao-Tsung and Jeng, Bingchiang(2005)'MFILM: a multi-dimensional fuzzy inductive learning method',Journal of Experimental & Theoretical Artificial Intelligence,17:3,267 — 281

To link to this Article: DOI: 10.1080/09528130500281828

URL: http://dx.doi.org/10.1080/09528130500281828

Journal of Experimental & Theoretical Artificial Intelligence,
Vol. 17, No. 3, September 2005, 267–281

Taylor & Francis
Taylor & Francis Group

MFILM: a multi-dimensional fuzzy inductive learning method

YAO-TSUNG CHEN*† and BINGCHIANG JENG‡

†Institute of Information Science, Academia Sinica, Taiwan
‡Department of Information Management,
National Sun Yat-sen University, Taiwan

(*Received 8 March 2005; revised 20 June 2005; accepted 22 June 2005*)

Inductive learning that creates a decision tree from a set of existing examples is shown to be useful for automated knowledge acquisition. Most of the existing methods however, handle only single-dimensional decision problems. Only some methods can deal with multi-dimensional decision problems. However, they are based on crisp concepts that are weak in handling marginal cases. In this paper, we present a multi-dimensional fuzzy inductive learning method that integrates the fuzzy set theory into the conventional multi-dimensional decision tree induction methods. The method converts a multi-dimensional decision tree into a fuzzy multi-dimensional 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 other conventional methods.

Keywords: Inductive learning; Machine learning; Expert systems; Multi-dimensional decision tree

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 (Braun and Chandler 1987), credit card application (Carter and Catlett 1987), graduate admission (Chung and Silver 1992), inventory accounting method choice (Liang *et al*. 1992), loan evaluation (Messier and Hansen 1988, Shaw and Gentry 1988) and medical diagnosis (Michalski *et al*. 1986, Liang *et al*. 1992). Their findings indicate that knowledge induced from these methods is equally or more accurate than statistical discriminant analysis (Fisher 1936) or other competing models in predicting new cases.

*Corresponding author. Email: ytchen@mail.iis.sinica.edu.tw

Journal of Experimental & Theoretical Artificial Intelligence
ISSN 0952–813X print/ISSN 1362–3079 online © 2005 Taylor & Francis
http://www.tandf.co.uk/journals
DOI: 10.1080/09528130500281828

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$30 000. That is, if the salary of an applicant is greater than US$30 000, then its credit is good. If the salary is less than or equal to US$30 000, then its credit classification is bad.

Following the segmentation of attributes, the discriminant power of each attribute is analysed. In a classic method called ID3 (Iterative Dichotomizer 3) (Quinlan 1979), the partition and discriminant power of an attribute are determined by a measurement called entropy gain. The attribute with a higher entropy gain is considered having a 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.

ID3 and other similar methods deal with problems with a single-dimensional class only, which can be called single-decision-tree-induction (SDTI) (Liu *et al*. 2000). In the real world, there are problems with a multi-dimensional class, so multi-decision-tree-induction (MDTI) methods are needed (Liu *et al*. 2000). For instance, in the car-buying decision problem, people may not consider which car but many properties of a car to buy, e.g. a car with moderate luxury, low price and high controllability. Methods can deal with this kind of problem may: (1) view the problem as many independent and separated problems with a single-dimensional class (Babič *et al*. 1999), (2) transform a multi-dimensional class into another single-dimensional class (Babič *et al*. 1999, Liu *et al*. 2000) or (3) integrate information measurements for each class dimension into a single measurement (Suzuki *et al*. 2001).

Although existing methods can deal with multi-dimensional decision problems, 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$30 000 annually is considered good, but another person making US$29 999 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 multi-dimensional 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 multi-dimensional fuzzy inductive learning method (MFILM) 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 MFILM can outperform than existing methods.

The remainder of the paper is organized as follows. Basic concept of the inductive learning process is introduced in section 2. Section 3 will introduce MDTI techniques. This is followed by section 4, which is an introduction to the fuzzy set concepts. Then, the MFILM is presented in section 5. Finally, empirical results and conclusions are presented in sections 6 and 7 respectively.

2. SDTI learning

Induction is a process by which a knowledge structure can be created from a set of data to explain or predict the behaviour of the data set. Early work of inductive learning can be traced back to 1966 when Hunt *et al*. (1996) developed a method for induction. This method was later implemented and expanded by Paterson and Niblett (1982) to create ACLS (A Concept Learning System) and by Quinlan (Quinlan 1979, 1986) to develop the popular ID3. The C4.5 algorithm (an extension of ID3) (Quinlan 1993) 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 gain of the decomposition,
     - Choose the value whose entropy gain is the largest,
   - 2.2. For each categorical attribute, decompose C by its classes and calculate its entropy gain,
   - 2.3. Choose the attribute F^* whose entropy gain 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_j to its leaves,
3. For $i=1$ to k, if cases in C_i are not of the same class, then let $C=C_i$ and go to 2, otherwise stop.

This method assumes that each example has a single-dimensional class, and can thus be called as SDTI (Liu *et al*. 2000). In dealing with real data, however, we often have a multi-dimensional class, and may wish to predict a multi-dimensional class (Caruana 1997a, 1997b). Therefore, MDTI methods are needed. The next section will introduce some existing MDTI methods.

3. MDTI learning

In MDTI, a set of cases is defined as $C=\{(v_{i1},\ldots, v_{in}; g_{i1},\ldots, g_{ik})|v_{ij}\in V_j$ where V_j is the domain of an attribute F_j and g_{ik} is the kth class dimension of case $i\}$. One possible learning process is to build a separate decision tree for each class dimension k independently (Babič *et al*. 1999). We will call this kind of method hereafter, Multiple Independent Decision Trees Induction (MIDTI). However, that would be problematic when there is a mutual dependence between class dimensions. Because attribute segmentation with independent measurements for each class

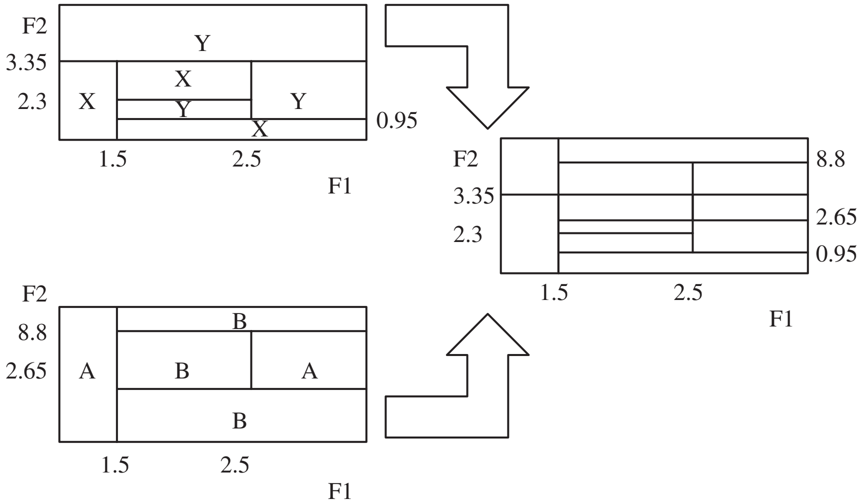

Figure 1. Space partition lapping.

Table 1. A data set of two attributes and two class dimensions.

No.	Class 1	Class 2	Transformed Class	F1	F2
1	X	A	P	1	3.2
2	X	A	P	1	2.5
3	X	A	P	1	1.5
4	X	B	Q	2	2.4
5	X	B	Q	2	3.1
6	Y	B	S	2	2.2
7	Y	B	S	3	1.1
8	Y	B	S	3	9.1
9	Y	A	R	3	3.5
10	Y	A	R	4	8.5
11	Y	A	R	4	2.9
12	X	B	Q	4	0.8

dimension ignores the mutual dependence, the resulting knowledge model will lack this relationship. This method can be view as lapping a partition to another partition in the attribute space. Figure 1 shows partitions of the attribute space induced from the data set listed in table 1. We can see there are too many subspaces, and this will affect the robustness of the model (i.e. overfitted).

Another possible learning process is to transform class dimensions into a new single-dimensional class and a conventional decision tree learning algorithm can be applied to construct a decision tree for the new class (Babič *et al.* 1999, Liu *et al.* 2000). This idea is described in (Caruana 1997a, 1997b) briefly without experimental justification. A transformation may be with or without loss of information. Assigning a new class value to each combination of class values is a transformation without loss of information. We will call this kind of method hereafter, Many-to-one Transformed Decision Tree Induction (MTDTI). The resulting partition is shown

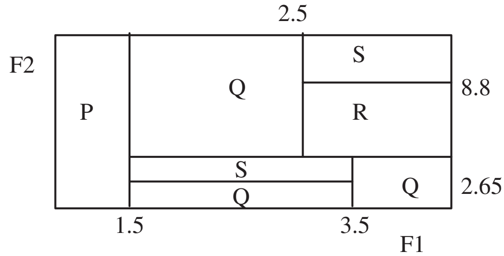

Figure 2. Space partition induced from the new class.

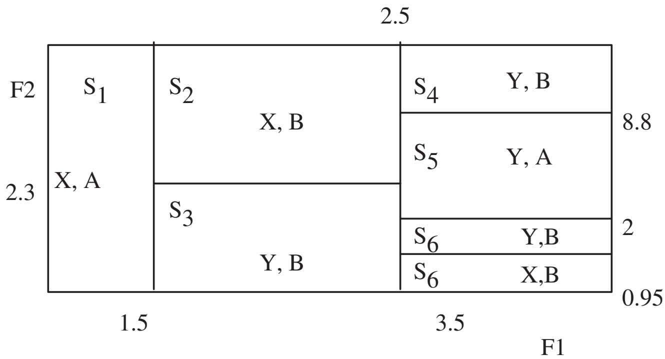

Figure 3. Space partition using add-sum gain ratios.

in figure 2. Here, we can find two attributes are selected with interleave and a complex partition is constructed, which is not what we want.

Besides, this transformation considerably increases the number of class values, which causes a fragmentation problem (Quinlan 1993): each class value has only a few training examples in a split node at the bottom of a decision tree, and appropriate selection of an attribute would be difficult. A method to avoid the fragmentation problem is not to construct a decision tree by partitioning directly. Babič *et al*. (1999) use evolutionary method instead of partitioning method, which start with the random-composed population of decision trees and let the evolution produce and select the final decision tree. Another method is to construct multi-dimensional decision trees, each of which corresponds to a particular decision class. To construct a decision tree for a respective decision class, the original training instances need to be re-assign into two revised decision classes, positive and negative (Liu *et al*. 2000). On the other hand, a transformation with loss of information such as principle component analysis (Kendall 1980, Forouraghi *et al*. 1994) could overlook useful knowledge.

The third possible learning process is to integrate information measures for each class dimension k into a single measurement and a conventional decision tree learning algorithm can be applied. We will call this kind of method hereafter, Many-to-one Metric Decision Tree Induction (MMDTI). Figure 3 shows the resulting space partition. We can find the partition shown in figure 3 is much simpler than those in figures 1 and 2. This can be attributed to the usage of complete information from all class dimensions.

Suzuki *et al.*'s work (2001) is an example of this kind of method, which use the add-sum of gain ratio (Quinlan 1986, 1993) for each class dimension k as the measurement and invent a new kind of decision tree called 'bloomy decision tree', which predicts g_{ik} in an internal node which is called a flower node. The construction method is similar to traditional one. The existence of 'flower node' is useful for pre-pruning; otherwise, all class dimensions can be predicted in leaf nodes.

Although Suzuki and *et al.*'s works (2001) have solved some of the problems, there is another issue that has not been addressed. That is their method can only apply to categorical attributes. In this paper, we will present a method which can apply both to categorical and continual attributes. However, we will not use crisp hurdle values to decompose the training data into subsets (e.g. C4.5), because in the real world, many decision boundaries are fuzzy and marginal differences are often ignored. A crisp multi-dimensional decision tree is easy to read and to follow. However, marginal cases are often misclassified due to the non-compensatory nature of the multi-dimensional decision tree (i.e. the weakness in one attribute cannot be offset by the strength in another). This indicates that the crisp hurdle value employed in traditional multi-dimensional inductive learning methods may cause problems and must be handled in different ways (Jeng and Jeng 1993, Jeng *et al.* 1997). In the following sections, we present a fuzzy approach to alleviate the above problem.

4. Fuzzy sets theory

The fuzzy set theory was developed by Zadeh (1965) in 1965 and later extended and applied to many fields, including artificial intelligence. For instance, a recent work by Jeng and Liang (1995) 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.

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:

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

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:

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

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:

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

Sometimes, the above definitions have problems in capturing all relevant information. For instance, the resulting membership degree of $A \cup B$ 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 $\mu_B(x)$.

Another approach to overcome the problem is to use a family of functions known as Yager class to modify the operations (Yager 1980, Klir and Floger 1988). Yager class uses a parameter w to control the strength of the resulting membership function, where $1 \leq w \leq \infty$. The union and intersection functions can be redefined as follows:

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

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

These two functions are general forms of our previous definitions. When $w = \infty$, the Yager functions of $\mu_{A \cup B}(x)$ and $\mu_{A \cup B}(x)$ become our original definitions, that is, $\max(\mu_A(x),\ \mu_B(x))$ and $\min(\mu_A(x),\ \mu_B(x))$ (Yager 1980, Klir and Floger 1988). Since Yager functions provide more flexibility, we employ them in our tree induction method.

5. Fuzzy MDTI

In some sense, the induction of multi-dimensional 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 multi-dimensional decision tree as shown in figure 4a partitions the space of two attributes, (X, Y), into three subspaces as shown in figure 4b. Subspaces S_a and S_c contain cases of class $(A, 1)$. Subspace S_b contains cases of class $(B, 2)$. In a traditional multi-dimensional 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 $(A, 1)$ no matter how small is the difference.

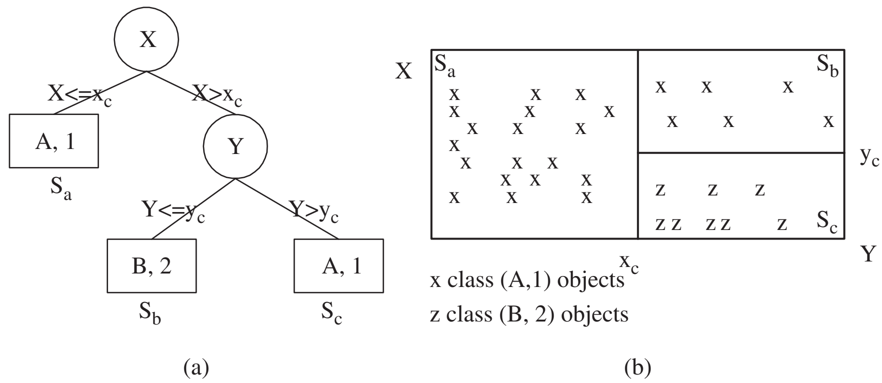

Figure 4. Space partition by a multi-dimensional decision tree: (a) a crisp multi-dimensional decision tree; (b) partition of attribute space.

The fuzzy MDTI process uses the crisp multi-dimensional decision tree generated from traditional MDTI 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 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 multi-dimensional decision tree becomes fuzzy at this stage. Therefore, a complete fuzzy multi-dimensional decision tree is composed of fuzzy hurdle values and fuzzy classes. When it is applied to analyse a new case, a defuzzification process must be used to convert fuzzy classes into a conclusion. In the following sections, we will describe the fuzzification and defuzzification processes in detail.

5.1. *Fuzzification of a hurdle value*

The fuzzification of a hurdle value changes the value from exactly x_c to about $\underline{x}_c$. We use x_c, to stand for the fuzzy value of x_c. Accordingly, the relationships of $x \le x_c$ and $x > x_c$ can also be changed to $x \le \underline{x}_c$ and $x > \underline{x}_c$, respectively.

The fuzzy membership functions of $x \le \underline{x}_c$, and $x > \underline{x}_c$, 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:

$$\mu_{x \le \underline{x}_c}(x) = \begin{cases} 1, & \text{if } x \le x_{cl} \\ (x_{cu} - x)/(x_{cu} - x_{cl}), & \text{if } x_{cl} < x \le x_{cu} \\ 0, & \text{if } x_{cu} < x \end{cases} \quad (5)$$

$$\mu_{x > \underline{x}_c}(x) = \begin{cases} 0, & \text{if } x \le x_{cl} \\ (x - x_{cl})/(x_{cu} - x_{cl}), & \text{if } x_{cl} < x \le x_{cu} \\ 1, & \text{if } x_{cu} < x \end{cases} \quad (6)$$

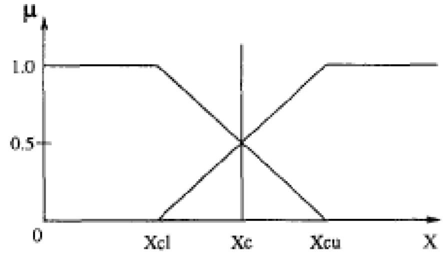

Figure 5. A fuzzy boundary.

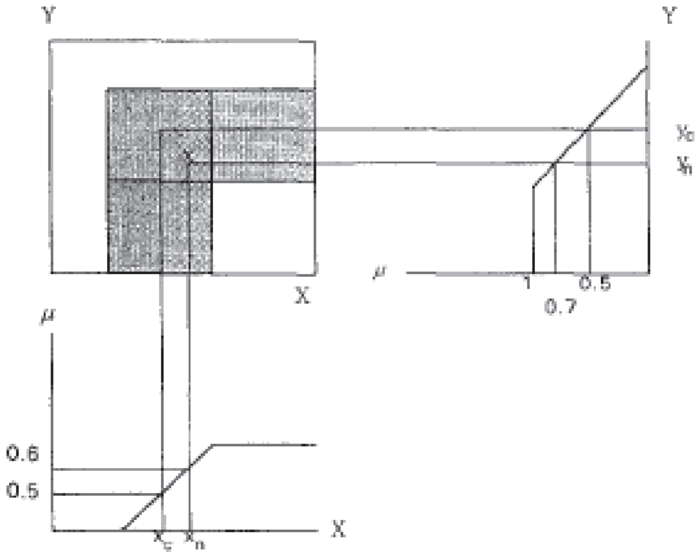

Figure 6. Fuzzy association of an object.

Figure 5 shows the graphical form of the membership functions. Suppose $x_{cl}=0$ and $x_{cu}=10$, then $\mu_{x \leq \underline{x}c}(x=8)=0.2$ and $\mu_{x > \underline{x}c}(x=8)=0.8$.

5.2. *Fuzzy classification of tree leaves*

Each leaf of a multi-dimensional decision tree is a subspace as shown in figure 4, 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 grey areas in figure 6. 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 $x > \underline{x}_c$ and $y \leq \underline{y}_c$,. That is,

$$\mu_{S_b}(x, y) = \mu_{x>\underline{x}c}(x) \cap \mu_{y\leq \underline{y}c}(y) \tag{7}$$

If two attributes X and Y are both fuzzified, then we can apply equation 2 or equation 4 to calculate the membership degree that an observation falls into a subspace. If we use equation 2 for simplicity, then a case having $\mu_{x > \underline{x}c}(x) = 0.6$ and $\mu_{y \leq \underline{y}c}(y) = 0.8$ 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.

5.3. *Construction of a fuzzy MDTI*

The final step of fuzzy MDTI is to fuzzify the crisp multi-dimensional decision tree so that the hurdle values and leaf classifications are both fuzzy. For example, the crisp multi-dimensional decision tree shown in figure 4a must be converted to the fuzzy multi-dimensional decision tree as shown in figure 7. This stage consists of two steps: reclassification of training cases and calculation of class memberships.

5.3.1. Reclassification of training cases. Since the crisp hurdle values have been replaced by fuzzy values, all training cases must be analysed using the procedures described in sections 5.1 and 5.2 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 equation 2 or Yager function in equation 4 can be used.

5.3.2. 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 figure 7 belongs to class $(A, 1)$ or $(B, 2)$]. 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 $(A, 1)$ or $(B, 2)$ (i.e., $\mu_{A,1}$ or $\mu_{B,2}$) is the union of all possibilities that the cases of class $(A, 1)$ or $(B, 2)$ falling in the leaf.

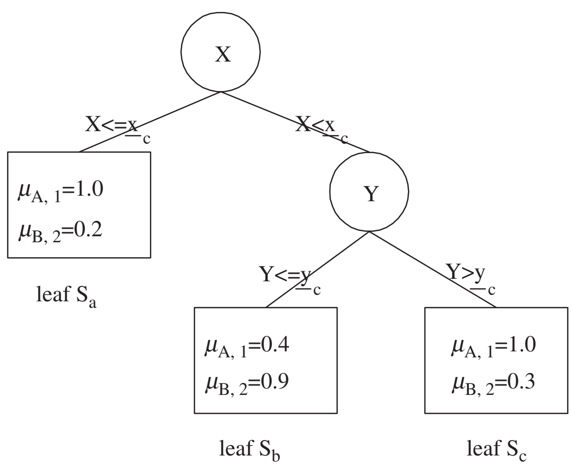

Figure 7. A fuzzy decision tree.

Formally, we can define as follows [$C_{a,1}$ and $C_{b,2}$ stand for all cases of classes $(A, 1)$ and $(B, 2)$ respectively]:

$$\mu_{A,1}(S_i) = \cup_{C_{A,1}} \mu S_i(x, y), \text{ for all cases } (x, y) \text{ in } C_{A,1} \quad (8)$$

$$\mu_{B,2}(Si) = \cup_{C_{B,2}} \mu S_i(x, y), \text{ for all cases } (x, y) \text{ in } C_{B,2} \quad (9)$$

5.4. *Prediction of new cases*

Since most information in a fuzzy multi-dimensional 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 5.3 regarding the reclassification of training cases. They will have different degrees of association with each leaf. 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. Finally, we need a defuzzification mechanism to conclude exactly which class the case belongs. The results obtained from the previous stage are often contradictory. Again, there are many different ways for defuzzification. The simplest one we use is called the total value sum approach. The approach summarizes possibilities of all leaves and the class with the highest sum is concluded to be the class of the new case.

Other criteria that may be used for defuzzification include K-sum (summarize highest K values of class association and choose the class with the highest value, value sum of k-nearest neighbours (summarize the possibilities of k nearest neighbours and choose the class with the highest total possibility value), majority class of K-nearest neighbours (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.

6. Empirical evaluation

In section 5, we have presented the process of MFILM. To further understand the performance of MFILM, we apply it to analyse three sets of data.

6.1. *Data sets*

The data sets employed for evaluation are obtained from UCI Repository (Blake *et al.* 1998). Their features are briefly described below and summarized in table 2.

1. Autos data: This set was used by Kibler *et al.* (1989) to predicted price of car using all numeric and boolean attributes. The set contains 205 cases. Each case

Table 2. Summary of training data sets.

Data sets	Class dimensions	Classes[a]	Categorical attributes	Quantitative attributes	Training samples	Testing samples	Total samples
Autos	2	7, 8 (12)	9	15	80	79	159 (205)[b]
	3	7, 8, 7 (19)	8				
	4	7, 8, 7, 2 (27)	7				
	5	7, 8, 7, 2, 3 (34)	6				
	6	7, 8, 7, 2, 3, 2 (47)	5				
Bridges	2	3, 2 (4)	5	2	42	29	71 (108)[b]
	3	3, 2, 3 (8)					
Flags	2	6, 8 (24)	17	10	100	94	194
	3	6, 8, 4 (38)	16				

[a]Numbers in parentheses indicate the number of multi-dimensional combination.
[b]Numbers in front of parentheses indicate the actual number of cases used in the experiments.

consists of 15 numerical and 11 categorical attributes. Several of the attributes in the database could be used as a 'class' attribute. The actual number of cases used was 159 because cases containing missing values were removed in our experiment.

2. Bridges data: Reich (1990) describes a new bridge design method using this set. It is a design domain where five design description need to be predicted based on seven specification properties. The original set contained 108 cases. After removing those with missing values, 71 were actually used in our experiment.
3. Flags data: This data set contains details of various nations and their flags and consists of 194 cases. Several attributes of a nation can be predicted from properties on its flag. Each case is represented by 10 numerical and 19 categorical attributes.

6.2. *Experimental procedures*

The treatment of the experiment is different induction methods. Four methods were compared: MIDTI, MTDTI, MMDTI and MFILM. In this experiment, MFILM is a post-treatment of traditional MMDTI methods. All methods are implemented in Borland C++ Builder. MIDTI is to construct a decision tree for each class dimension independently by C4.5. MTDTI is to apply C4.5 to the new class dimension which is transformed from a multi-dimensional class by assigning a new class value to each combination of class values. MMDTI is to apply C4.5 to the original multi-dimensional class with a single integrated measurement. All method use gain ratio as the measurement. We use MIDTI, MTDTI and MMDTI as the benchmark for evaluating MFILM.

Since the defuzzication procedure we chosen may produce more than one predictions, we use *precision* of a prediction, instead of accuracy, to compare the performance of these methods. The *precision* of a prediction was measured using the hold-out sample. That is, we select a part of cases randomly as training data and the induced tree is then applied to predict the hold-out testing cases. The *precision* of a prediction is defined as the fraction of a prediction that matches all class dimensions, and the *precision* is either 0 (for a miss) or 1 divided by the number of prediction classes (for a hit). For each individual prediction, the *precision* is a value in [0, 1].

For Bridges, there are already class dimensions. For others, we choose some categorical attributes (not binary) as class dimensions. For each class dimensions in each set, a SDTI task is settled, and class dimensions with which C4.5's accuracy is sorted. At lease three highest-accuracy class dimensions plus those whose accuracy are greater than 50% will be selected to conduct the following experiments. For each data set, we first use two highest accuracy class dimensions to settle a MDTI task, and add one class dimensions each time until all selected class dimensions are used.

We control two parameters of MFILM during the experiment: Yager's w and the width of the fuzzy border (i.e. $x_{cu} - x_{cl}$). We experimented with different w values ranging from 1 to 8 stepping 1. The width of the fuzzy border was determined by the standard deviation of the partitioned samples. We experimented with widths from 0.25 to 1 standard deviation stepping 0.25.

6.3. *Results and discussions*

Table 3 shows the predictive precision of each method when applied to different data sets with different numbers of class dimensions. The Yager's w and fuzzy width are the parameter values we actually used in the experiment to achieve the precision. The average predictive precisions are 45.72% for MIDTI, 55.31% for MTDTI, 55.40% for MMDTI and 58.17% for MFILM. The result is obvious: MFILM outperformed all three other methods. A single-tail t-test indicates that the difference is statistically significant at 10% level ($\alpha = 0.1$, $t = 1.480$) between MFILM and MIDTI. Therefore, we can conclude that MFILM is significantly better than MIDTI.

MFILM is a post-treatment of traditional MDTI methods. In this section, we have seen that original MDTI methods can be improved in its predictive

Table 3. Prediction accuracy of different methods in data sets with quantitative attributes.

Data sets	MIDTI	MTDTI	MMDTI	MFILM	Yager's w	Fuzzy width
Autos-2	70.21	83.54	86.08	**89.87**	4	0.5/1
Autos-3	61.70	**81.01**	65.82	69.62	1	0.75
Autos-4	52.13	**72.15**	64.56	65.82	3∼5	0.25
Autos-5	46.81	60.76	**62.03**	**62.03**	2∼6	0.25
					6	0.5
Autos-6	34.04	45.57	49.37	**51.90**	4	0.75
Bridges-2	58.62	62.07	62.07	**68.97**	5∼7	0.25
Bridges-3	37.93	31.03	44.83	**48.28**	4∼8	0.5
					6∼8	0.75
					8	1
Flags-2	35.11	38.30	37.23	**40.43**	1/2	0.25
Flags-3	14.89	23.40	26.60	**26.60**	1∼8	0.25∼1
Average	45.72	55.31	55.40	58.17		

precision after processing by MFILM. However, MFILM 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 width. Based on the results, it seems that best Yager's *w* and best fuzzy width for each data set is very different.

The reason that the fuzzy tree generated by MFILM was more accurate than the original tree generated by other methods 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.

7. Conclusions

Tree induction has been a major technique for automated knowledge acquisition. Conventional learning algorithms are ineffective in constructing an accurate decision tree in a multi-dimensional class. Moreover, most existing MDTI methods are crisp in nature. They create multi-dimensional 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 MFILM outperformed the MIDTI approach significantly.

Although our results show the potential of MFILM, there are some issues that require further research. First, 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. Second, a good MDTI 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 MFILM and other methods.

Acknowledgements

This research was supported in part by the National Science Council of Taiwan under contracts NSC89-2416-H-110-088 and NSC93-2524-S-001-001.

References

Š.H. Babič, M. Šprogar, M. Zorman, P. Kokol and D.M. Turk, "Evaluating breastfeeding advantages using decision tree", in *Proceeding of the 12th IEEE Symposium on Computer-Based Medical Systems*, Stanford, CT, 1999, pp. 144–149.

C. Blake, E. Keogh and C.J. Merz, "UCI repository of machine learning databases", Department of Information and Computer Science, University of California, Irvine. Available online: http://www.ics.uci.edu/~mlearn/MLRepository.html

H. Braun and J.S. Chandler, ''Predicting stock market behavior through rule induction: an application of the learning-from-example approach'', *Decision Science*, 18, pp. 415–429, 1987.
C. Carter and J. Catlett, ''Assessing credit card applications using machine learning'', *IEEE Expert*, 2, pp. 71–79, 1987.
R. Caruana, ''Multitask learning'', PhD Thesis, CMU-CS-97-203, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA (1997a).
R. Caruana, ''Multitask learning'', *Machine Learning*, 28, pp. 41–75, July 1997b.
H.M. Chung and M.S. Silver, ''Rule-based expert systems and linear models: an empirical comparison of learning-by-examples methods'', *Decision Science*, 23, pp. 687–707, 1992.
R.A. Fisher, ''The use of multiple measurements in taxonomic problems'', *Annals of Eugenics*, 7, pp. 179–188, 1936.
B. Forouraghi, L.W. Schmerr and G.M. Prabhu, ''Induction of multivariate regression trees for design optimization'', in *Proceeding of 12th National Conference on Artificial Intellignece*, Seattle, WA, 1994, pp. 607–612.
E.B. Hunt, J. Marin and P.T. Stone, *Experiments in Induction*, New York: Academic Press, 1996.
B. Jeng and Y.-M. Jeng, ''A fuzzy tree induction learning method'', in *Proceeding of the First Asian Fuzzy Systems Symposium*, Singapore, 1993.
B. Jeng, Y.-M. Jeng and T.-P. Liang, ''FILM: a fuzzy inductive learning method for automated knowledge acquisition'', *Decision Support System*, 21, pp. 61–73, 1997.
B. Jeng and T.-P. Liang, ''Fuzzy indexing and retrieval in case-based systems'', *Expert Systems with Applications*, 8, pp. 135–142, 1995.
M.G. Kendall, *Multivariate Analysis Second Edition*, High Wycombe: Charles Griffin, 1980.
D. Kibler, D.W. Aha and M. Albert, ''Instance-based prediction of real-valued attributes'', *Computational Intelligence*, 5, pp. 51–57, 1989.
G.J. Klir and T.A. Floger, *Fuzzy Sets, Uncertainty, and Information*, Englewood Cliffs, NJ: Prentice-Hall, 1988.
T.-P. Liang, J.S. Chandler and I. Han, ''Integrating statistical and inductive learning methods for knowledge acquisition'', *Expert Systems with Applications*, 1, pp. 391–401, 1990.
T.-P. Liang, J.S. Chandler, I. Han and R. Roan, ''An empirical investigation of some data effects on the classification accuracy of probit, ID3 and neural networks'', *Contemporary Accounting Research*, 9, pp. 306–328, 1992.
S.O.R. Liu, C.-P. Wei, P.J.-H. Hu and N. Chang, ''Automated learning of patient image retrieval knowledge: neural networks versus inductive decision trees'', *Decision Support Systems*, 30, pp. 105–124, 2000.
W.F. Messier Jr and J.V. Hansen, ''Inducing rules for expert system development: an example using default and bankruptcy data'', *Management Science*, 34, pp. 1403–1415, 1988.
R. Michalski, I. Mozetic, J. Hong and N. Lavrac, ''The multi-purpose incremental learning system AQl5 and its testing application to three medical domains'', in *Proceedings of the 5th National Conference on Artificial Intelligence*, Philadelphia, PA, 1986, pp. 1041–1045.
A. Paterson and T. Niblett, *ACLS User Manual*, Glasgow: Intelligence Terminal, 1982.
J.R. Quinlan, ''Discovering rules from large collections of examples a case study'', in *Expert Systems in the Micro Electronic Age*, D. Michie, Ed., Edinburgh: Edinburgh University Press, 1979.
J.R. Quinlan, ''Induction of decision trees'', *Machine Learning*, 1, pp. 81–106, 1986.
J.R. Quinlan, *C4.5: programs for machine learning*, San Mateo, CA: Morgan Kaufmann, 1993.
Y. Reich, ''Converging to 'ideal' design knowledge by learning'', in *Proceedings of the First International Workshop on Formal Methods in Engineering Design*, Colorado Springs, CO, 1990, pp. 330–349.
M.J. Shaw and J.A. Gentry, ''Using an expert system with inductive learning to evaluate business loans'', *Financial Management*, 17, pp. 45–56, 1988.
E. Suzuki, M. Gotoh and Y. Choki, ''Bloomy decision tree for multi-objective classification'', *Lecture Notes in Artificial Intelligence*, 2168, pp. 436–447, 2001.
J. Wnek, J. Sarma, A. Wahab, and R. Michalski, ''Comparison learning paradigms via diagrammatic visualization: a case study in single concept learning using symbolic neural net and genetic algorithm methods'', Technical Report, Computer Science Department, George Mason University, 1990.
R. Yager, ''On a general class of fuzzy connectives'', *Fuzzy Sets and Systems*, 4, pp. 235–242, 1980.
L.A. Zadeh, ''Fuzzy sets'', *Information and Control*, 8, pp. 338–353, 1965.