Baldwin, J. F., Lawry, J., & Martin, T. P. (1997). Mass assignment fuzzy ID3 with applications. In In Fuzzy logic - applications and future directions (Unicom Seminars). (pp. 278 - 294)

Mass Assignment Fuzzy ID3 with Applications

J.F. Baldwin, J.Lawry[*] and T.P. Martin

A.I. Group,
Department of Engineering Mathematics,
University of Bristol, Bristol BS8 1TR
United Kingdom

Abstract
A mass assignment based ID3 algorithm for learning probabilistic fuzzy decision trees is introduced. Fuzzy partitions are used to discretise continuous feature universes and to reduce complexity when universes are discrete but with large cardinalities. Furthermore, the fuzzy partitioning of classification universes facilitates the use of these decision trees in function approximation problems. The potential of this approach is then illustrated by its application to a number of test and real world problems.

Keywords: mass assignment, fuzzy partition, fuzzy probabilistic decision tree.

1 Introduction

Over the last few decades there has been a wide spread tendency throughout society to collect large amounts of data relating to almost any issue for which some form of quantitative analysis is possible. Implicit in such data is information on patterns and relationships holding between the particular measured features which needs to be extracted if this wealth of material is to be fully exploited. The objective then is to develop methods for generating rules which express this information.

The ID3 algorithm introduced by Quinlan [15] has proved to be an effective and popular method for finding decision tree rules to express information contained implicitly in discrete valued data sets. There are, however, a number of well known difficulties associated with the application of this method to real world problems. For instance, the decision tree generated is equivalent to a set of first order logic conditionals each of which is true for every element of the data set. In other words, if we generate classification rules relating a set of classes to values of some set of attributes then correct classification is guaranteed for each element in the training set. A natural consequence of this property is that classical ID3 is inappropriate for databases containing significant noise since the generated rules will then fit the noise and this may lead to a high error rate when classifying unseen cases. Furthermore, often in practice classification problems have continuous attribute values associated with them necessitating the partitioning of relevant universes if ID3 is to be applied. This is essentially the approach adopted in the C4.5 algorithm [17], a successor to ID3 where the universe of a continuous attribute $\boxed{A}$ is partitioned by the two sets $A > \alpha$ and $A \leq \alpha$ for some parameter α. The use of crisp partitions in this case can be problematic since sudden and inappropriate changes to the assigned class may result from small changes in attribute values. Clearly such behaviour will reduce the generalisation capabilities of the system. A further limitation is the inability to utilise or classify data points where some of the attribute values have not been specified although in the C4.5 algorithm this problem is partially overcome by exploring all possible branches of the tree consistent with this point and then combining the results. Finally, the requirement that the set of classification values be finite and mutually exclusive means that classical ID3 and C4.5 cannot be applied to more general problems such as

[*] Supported by an E.P.S.R.C research assistantship

function approximation or the generation of rules to summarise information stored in large databases.

The use of fuzzy sets to partition universes can have significant advantages over more traditional approaches and when combined with classical decision tree induction methods can help to address many of the difficulties discussed above. In particular, fuzzy decision rules tend to be more robust and less sensitive to small changes in attribute values near partition boundaries. Also such rules will tend to have greater generalisation capabilities than their crisp counterparts since the requirement of one hundred percent correct classification of the training set has been relaxed. The concept of fuzzy partition (see [18]) allows us to incorporate both overlapping classification and attribute classes into our induction model in a coherent way. This can have advantages in terms of tree complexity since empirical evidence suggests that this less restrictive notion of partition enables fewer attribute classes to be used. In addition, many problems can best be expressed using concepts most naturally corresponding to overlapping classes and hence in this sense fuzzy partitions can facilitate the generation of rules more easily understood by humans. Of course, there is another way in which the incorporation of fuzzy sets can produce more ‘human’ decision rules since they are able to model vague concepts such as those found in natural language. This is particularly useful in the case of continuous variables where it can be helpful to give linguistic labels to the fuzzy sets such as, for example, high, medium and low. A further advantage in fuzzy partitions is that the inherent interpolation properties of smooth fuzzy sets enables the decision tree to be used, in conjunction with a defuzzification method, for function approximation.

In the sequel we describe a method for generating probabilistic decision trees with fuzzy attribute and classification values. The decision trees generated are probabilistic classifiers analogous to those suggested by Quinlan in [16] where the probabilities are calculated according to the mass assignment semantics for fuzzy sets developed by Baldwin (see [3] and [4]). This algorithm has been implemented in Fril [1] which is a logic programming style language with built in capabilities for processing both probabilistic and fuzzy uncertainty. The decision trees can be represented in terms of Fril extended rules the syntax and semantics of which will be described in a later section.

2 The Notion of Fuzzy Partitions

In this section we introduce the basic idea of a fuzzy partition and describe how such partitions are utilised in the fuzzy ID3 algorithm. The notion of a partition of a universe has been extended to fuzzy sets by Ruspini [18] as follows:

Definition 2.1

The set of fuzzy sets $\left\{f_1,\cdots,f_n\right\}$ form a **fuzzy partition** of the universe Ω iff

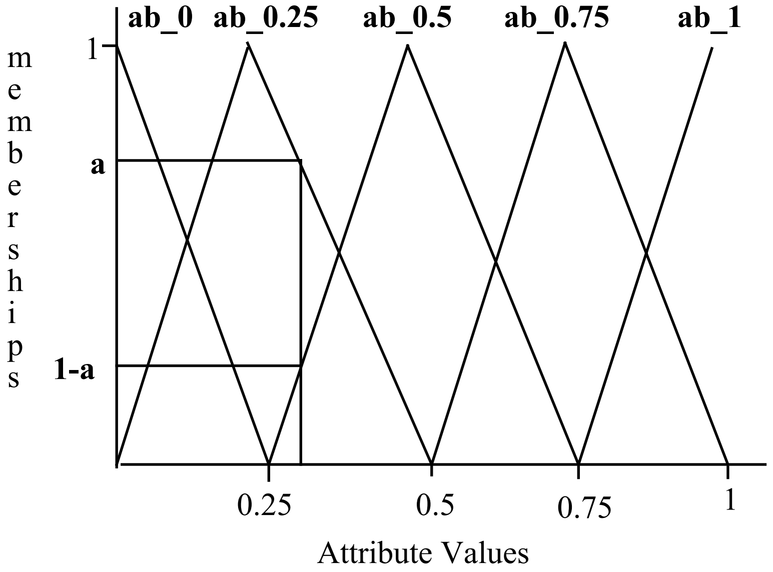
$$\forall x \in \Omega \sum_{i=1}^{n} \chi_{f_i}(x) = 1$$

The essential requirement, then, is that the sum of the membership values for an element of the universe across the partition is one. Furthermore, notice that if $f_1,\cdots,f_n$ are restricted to crisp sets then this corresponds to the standard definition for a partition of Ω.

Figure 1 shows an example of a fuzzy partition of [0, 1] consisting of triangular fuzzy sets where each member of the partition can be viewed as a fuzzy or imprecise value for an attribute. In fact such simple fuzzy sets have been found to be extremely effective in many applications.

The need to partition universes introduces a new problem into decision tree induction; namely how to decide on the exact form of a partition for any given variable. In the current context this problem is naturally divided in two distinct sub-problems. These are the partitioning of universes of classification values and the partitioning of attribute universes. For classification universes we use an algorithm originally developed for the Fril data browser [5] based on the heuristic that the classification values generated by the data set should be evenly distributed across the partition sets. A number of partition points are selected on this basis each of which forms the apex of a triangular fuzzy set constructed so that together they form a fuzzy partition in accordance with definition 2.0.1. Notice that some user involvement is still required, however, since the number of fuzzy sets in the partition must be specified.

Figure 1



With regard to partitioning attribute universes we have, for the test cases presented in the following sections, adopted a fairly simple minded approach and used fuzzy partitions consisting of evenly spaced triangular fuzzy sets, again where the actual number of sets required is specified by the user. There are of course many more sophisticated partitioning techniques that could be considered here and in particular the use of clustering techniques to generate apex points for triangular partitions might seem worthy of consideration. Empirical testing has suggested, however, that such approaches rarely improve on results obtained with a uniform partition and in certain cases can even lead to a deterioration in performance.

3 Matching Fuzzy Sets

In order to generate the probability values required for the decision tree induction a method is required for determining the level at which two fuzzy sets match. In particular, to determine the level of support afforded to a fuzzy clause or statement of the form A is $\boldsymbol{f}$, where A is an attribute and $\boldsymbol{f}$ is a fuzzy value of A, by a data object, say o, we need to be able to evaluate a conditional probability (or support) for $\boldsymbol{f}$ given o_A, the object's value for A.

The notion of semantic unification developed for the programming language Fril [1] provides just such a mechanism. Semantic unification is based on an alternative definition of the conditional probability of fuzzy sets extending Zadeh's original definition of the probability of fuzzy set (see [20]).

Definition 3.1

General Case:

Let $\boldsymbol{f}$ and $\boldsymbol{g}$ be fuzzy subsets of Ω and $\boldsymbol{P}$ be a probability distribution on Ω then

$$\boldsymbol{Prob_P(f| g)}=\int_0^1\int_0^1 \frac{\boldsymbol{P}(\boldsymbol{f}_{\mathrm{y}}\cap \boldsymbol{g}_{\mathrm{s}})}{\boldsymbol{P}(\boldsymbol{g}_{\mathrm{s}})}\,\mathrm{dsdy}$$

provided this integral exists and is undefined otherwise.

Discrete Case:

When Ω is finite the above definition can be expressed in terms of mass assignments as follows:

$$\boldsymbol{Prob_P}(\boldsymbol{f}|\boldsymbol{g})=\sum_{\mathrm{F_i}}\sum_{\mathrm{G_j}}\boldsymbol{P}\left(\mathrm{F_i}|\mathrm{G_j}\right)\boldsymbol{m_f}\left(\mathrm{F_i}\right)\boldsymbol{m_g}\left(\mathrm{G_j}\right)$$

where $\boldsymbol{m_f}, \{\mathrm{F_i}\}$ and $\boldsymbol{m_g}, \{\mathrm{G_i}\}$ are the mass assignment and set of focal elements for $\boldsymbol{f}$ and $\boldsymbol{g}$ respectively. See [3] for an introduction to mass assignment theory and for more details regarding the probability of fuzzy events see [4], [6], [7] and [20].

In order to illustrate this concept consider the elementary dice example below:

Example 3.2

Consider a fair six sided dice so that the probability distribution on {1, 2, 3, 4, 5, 6} is given by $\boldsymbol{P}(1)=\cdots=\boldsymbol{P}(6)=\frac{1}{6}$. Now suppose we know that the outcome of a throw of the dice is a $\boldsymbol{small\_value}$ where $\boldsymbol{small\_value}=1/\ 1+2/\ 0.7+3/\ 0.3$ and we want to know the probability that the outcome is $\boldsymbol{about\_two}$ where $\boldsymbol{about\_two}=1/\ 0.5+2/\ 1+3/\ 0.5$. Clearly then we must calculate $\boldsymbol{Prob_P}(\boldsymbol{about\_two}|\boldsymbol{small\_value})$. Now $\boldsymbol{m_{small\_value}}=\{1,2,3\}{:}0.3,\{1,2\}{:}0.4,\{1\}{:}0.3$ and $\boldsymbol{m_{about\_two}}=\{1,2,3\}{:}0.5,\{2\}{:}0.5$ so that the value for $\boldsymbol{Prob_P}(\boldsymbol{about\_two}|\boldsymbol{small\_value})$ can be determined with the aid of the following tableau.

m_{small_value}

		$\{1,2,3\}:0.3$	$\{1.2\}:0.4$	$\{1\}:0.3$
m_{about_two}	$\{1,2,3\}:0.5$	$P(\{1,2,3\}\|\{1,2,3\})=1$ $0.5\times 0.3=0.15$	$P(\{1,2,3\}\|\{1,2\})=1$ $0.5\times 0.4=0.2$	$P(\{1,2,3\}\|\{1\})=1$ $0.5\times 0.3=0.15$
	$\{2\}:0.5$	$P(\{2\}\|\{1,2,3\})=\frac{1}{3}$ $0.5\times 0.3=0.15$	$P(\{2\}\|\{1,2\})=\frac{1}{2}$ $0.5\times 0.4=0.2$	$P(\{2\}\|\{1\})=0$ $0.5\times 0.3=0.15$

From this we obtain

$$\boldsymbol{Prob_P}(\boldsymbol{about\_two}|\boldsymbol{small\_value})=1(0.15)+1(0.2)+1(0.15)+\frac{1}{3}(0.15)+\frac{1}{2}(0.2)=0.65$$

4 Fuzzy Probabilistic Decision Trees

We are now able to utilise the above ideas in order to develop a method to generate fuzzy decision trees from data. The trees induced will be probabilistic classifiers similar to those discussed in [16] although the method for obtaining the necessary probabilities is clearly quite different. The nodes will consist of attributes and each emergent branch will correspond to a fuzzy restriction on that attribute taken from a predefined fuzzy partition of its universe. In addition, the possibility of fuzzy classifications necessitates the incorporation of some form of defuzzification procedure into our system. In this section we shall describe the induction algorithm together with methods for classifying unseen cases in some detail.

Initially fuzzy partitions of all attribute universes with infinite or large cardinality are formed. For the attribute representing classification values the method described in section 2 is used to form a partition of triangular fuzzy sets over which there is a uniform spread of data classification values. Again as stated in section 2 the independent attributes are partitioned using evenly spaced triangular fuzzy sets although more sophisticated methods could be used here. In both cases the user is required to specify the number of fuzzy sets in the partitions.

Here and in the sequel we consider databases of the form

$$D=\left\{o_i=\left\langle o_{i,1},\cdots,o_{i,n}\right\rangle \middle| i=1,\cdots,N\right\}$$

where either $o_{i,j}$ is a value of the attribute A_j (i.e. $o_{i,j}\in\Omega_j$ where Ω_j is the universe of A_j) or $o_{i,j}$ is a fuzzy value of the attribute A_j (i.e. $o_{i,j}\subseteq_f \Omega_j$). Note that by allowing fuzzy values for attributes we are able to represent examples where some of the attribute values are unspecified, imprecisely specified, or vaguely specified. Now suppose that the fuzzy partition of Ω_j is $\mathbf{P}_j$ for $j=1,\cdots,n$ then D naturally generates a support for any compound statement of the form $B \equiv A_{i_1}$ is $\boldsymbol{f}_{i_1} \wedge\cdots\wedge A_{i_k}$ is $\boldsymbol{f}_{i_k}$ for $k \leq n$ and $\boldsymbol{f}_{i_r} \in \mathbf{P}_{i_r}$, proportional to the sum of products $w(B)=\sum_{t=1}^{N}\prod_{r=1}^{k}\boldsymbol{Prob}_{\boldsymbol{U}_{i_r}}\left(\boldsymbol{f}_{i_r}\middle| o_{t,i_r}\right)$ where $\boldsymbol{U}_j$ denotes the uniform measure on Ω_j. Statements of the above form characterise branches of fuzzy decision trees and hence we can utilise these supports in the learning process. In particular, to evaluate the conditional probability of A_{i_1} is $\boldsymbol{f}_{i_1}$ given A_{i_2} is $\boldsymbol{f}_{i_2} \wedge\cdots\wedge A_{i_k}$ is $\boldsymbol{f}_{i_k}$ we multiply $w(B)$ by an appropriate normalising constant. More specifically

$$\boldsymbol{Prob}\left(A_{i_1} \text{ is } \boldsymbol{f}_{i_1}\middle| A_{i_2} \text{ is } \boldsymbol{f}_{i_2} \wedge\cdots\wedge A_{i_k} \text{ is } \boldsymbol{f}_{i_k}\right)=\frac{w\left(A_{i_1} \text{ is } \boldsymbol{f}_{i_1} \wedge\cdots\wedge A_{i_k} \text{ is } \boldsymbol{f}_{i_k}\right)}{\sum_{\boldsymbol{f} \in \mathbf{P}_{i_1}} w\left(A_{i_1} \text{ is } \boldsymbol{f} \wedge\cdots\wedge A_{i_k} \text{ is } \boldsymbol{f}_{i_k}\right)}$$

A more detailed exposition of this method of calculating conditional probabilities from a database can be found in [9].

Conditional probabilities of the above form enable us to determine the expected information gain from evaluating an attribute given a particular branch B. The attribute which maximises this gain can then be select to extend the tree along B. In practice, we need only evaluate the expected entropy for each candidate attribute since the attribute with the lowest expected entropy will maximise the information gain. The expected entropy from evaluating attribute A, not appearing in B, is given by

$$I(A|B)=\sum_{\boldsymbol{f} \in \boldsymbol{P}_A} I(A \text{ is} \boldsymbol{f} \wedge B)\boldsymbol{Prob}(A \text{ is} \boldsymbol{f}|B)$$

where for any branch B

$$I(B)=-\sum_{\boldsymbol{f} \in \boldsymbol{P}_{\text{Class}}} \boldsymbol{Prob}(\text{Class is} \boldsymbol{f}|B)\boldsymbol{log}(\boldsymbol{Prob}(\text{Class is} \boldsymbol{f}|B))$$

The general algorithm for generating a fuzzy probabilistic decision tree from D given a set of fuzzy partitions and stopping thresholds is, therefore, as follows:

(1) For each branch B determine the maximum value of $\boldsymbol{Prob}(\text{Class is} \boldsymbol{f}|B)$ for $\boldsymbol{f} \in \boldsymbol{P}_{\text{Class}}$. If this is greater than a predefined threshold or B contains all available attributes then terminate B and quantify this branch with the distribution $\boldsymbol{Prob}(\text{Class is} \boldsymbol{f}|B)$. Otherwise go to (2)

(2) For every attribute A not occurring in B evaluate $I(A|B)$ and select the attribute A^* with the smallest value .

(3) Extend the tree by generating the new branches $B \wedge (A^* \text{is} \boldsymbol{f})$ for every $\boldsymbol{f} \in \boldsymbol{P}_{A^*}$ and go to (1).

Example 4.1
Consider a game played by between 1 and 8 people which simply involves each participant throwing a dice the winner being the individual with the highest score. In the case where more than one person has the highest score then each of them records a joint win. Suppose

the game has been played repeatedly over an evening and the results of a single individual have been recorded in the following database where the attributes are, from left to right, Outcome and Score and number of players.

D={<lose, 1/1+2/0.3, 8> ,
<joint_win, 4, 5 >,
<win, 6, 3>,
<win, 6/1 +5/0.6 +4/0.2, 4>
<lose, 4, 6>
<lose, 3/1+4/0.6, 2>
<joint_win, 5, 6>
<joint_win, 6, 4>}

We now partition the outcome universe {win, joint_win, lose} by ***success*** =win/1 +joint_win /0.6 , ***failure*** = lose/1 +joint_win/0.4, the score universe {1, 2, 3, 4, 5, 6} by ***high_score*** = 6/1 +5/0.8+4/0.3 and ***low_score*** = 1/1 +2/1 +3/1 +4/0.7 +5/0.2 and the player universe {1, 2, 3, 4, 5, 6, 7, 8} by ***many*** = 8/1 +7/1 +6/1 +5/0.5 +4/0.2 and ***few*** = 1/1 +2/1 +3/1 +4/0.8 +5/0.5. Suppose then we want to generate a decision tree to classify Outcome in terms of Score and Players. To make the initial choice of attributes we must first calculate the conditional distributions $\boldsymbol{Prob}(\text{Outcome}|\text{Score})$ and $\boldsymbol{Prob}(\text{Outcome}|\text{Players})$. The latter, for example, can be determined by summing the product $\boldsymbol{Prob}(\text{Outcome}|\text{o}_{\text{Outcome}})\boldsymbol{Prob}(\text{Players}|\text{o}_{\text{Players}})$ for the data points and then normalising across Outcome.

Outcome$|$***many***

success	***failure***
(0)(1) =0	(1)(1) =1
(0.6)(0.5) =0.3	(0.4)(0.5) =0.2
(1)(0) =0	(0)(0) =0
(1)(0.2)..=0.2	(0)(0.2) =0
(0)(1) =0	(1)(1) =1
(0)(0) =0	(1)(0) =0
(0.6)(1)..=0.6	(0.4)(1) =0.4
(0.6)(0.2) =0.12	(0.4)(0.2) =0.08
--------	-------
w_1 = 1.22	w_2 =2.68

$\boldsymbol{Prob}(\boldsymbol{success}|\boldsymbol{many})$=0.3128

$\boldsymbol{Prob}(\boldsymbol{failure}|\boldsymbol{many})$=0.6872

Outcome$|$***few***

success	***failure***
(0)(0) =0	(1)(0) =0
(0.6)(0.5) =0.3	(0.4)(0.5) =0.2
(1)(1) =1	(0)(1) =0
(1)(0.8) =0.8	(0)(0.8) =0
(0)(0) =0	(1)(0) =0
(0)(1) =0	(1)(1) =1
(0.6)(0) =0	(0.6)(0) =0
(0.6)(0.8) =0.48	(0.4)(0.8) =0.32
--------	-------
w_1 = 2.58	w_2 = 1.52

$\boldsymbol{Prob}(\boldsymbol{success}|\boldsymbol{few})$=0.6293

$\boldsymbol{Prob}(\boldsymbol{failure}|\boldsymbol{few})$=0.3707

Also it is found that $\boldsymbol{Prob}(\boldsymbol{few})$=0.5125 and $\boldsymbol{Prob}(\boldsymbol{many})$=0.4875
Hence we obtain

$$\mathrm{I}(\text{Players}) = 0.5125\left(-0.6293\,\boldsymbol{log}_2\,0.6293 - 0.3707\,\boldsymbol{log}_2\,0.3707\right)$$
$$+0.4875\left(-0.3128\,\boldsymbol{log}_2\,0.3128 - 0.6872\,\boldsymbol{log}_2\,0.6872\right) = 0.924849$$

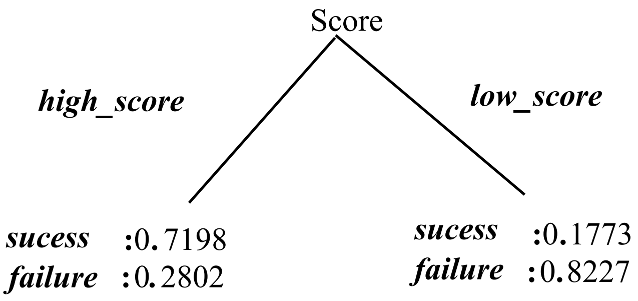
Similarly we find the relevant probabilities for Score to be

$\boldsymbol{Prob}(\boldsymbol{success}|\boldsymbol{high\_score}) = 0.7198$, $\boldsymbol{Prob}(\boldsymbol{failure}|\boldsymbol{high\_score}) = 0.2802$,

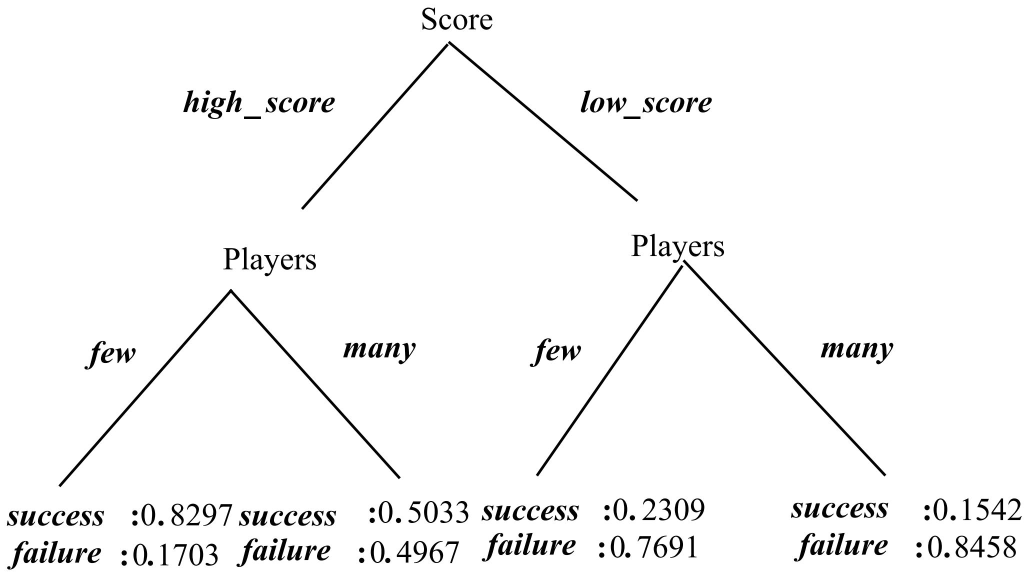
$\boldsymbol{Prob}(\boldsymbol{success}|\boldsymbol{low\_score}) = 0.1773$, $\boldsymbol{Prob}(\boldsymbol{failure}|\boldsymbol{low\_score}) = 0.8227$,

$\boldsymbol{Prob}(\boldsymbol{high\_score}) = 0.54875$ and $\boldsymbol{Prob}(\boldsymbol{low\_score}) = 0.45125$ giving
$\mathrm{I}(\text{Score}) = 0.773548$

Hence the attribute Score is selected to generate the following sub-tree:



Now setting the stopping threshold to 0.9 both branches fail to satisfy this criterion and hence we evaluate the remaining attribute Players to give:



For any decision tree of the form described above the branches correspond to a set of mutually exclusive and exhaustive events. This observation enables us to use probabilistic updating methods to determine the probability that a previously unseen example belongs to a particular class. More specifically, given a decision tree with branches $B_1,\cdots,B_T$ and test example $o=\left\langle o_1,\cdots,o_n\right\rangle$ an updated value for the probability of each classification can be found using Jeffrey's rule (see [13]) as follows:

$$\textbf{\textit{Prob}}\left(\text{Class is }\boldsymbol{f}|o\right)=\sum_{i=1}^{T}\textbf{\textit{Prob}}\left(\text{Class is }\boldsymbol{f}|B_i\right)\textbf{\textit{Prob}}\left(B_i|o\right)$$

Here the conditional probabilities $\textbf{\textit{Prob}}\left(\text{Class is }\boldsymbol{f}|B_i\right)$ are specified in the decision tree and

$$\textbf{\textit{Prob}}\left(B|o\right)=\prod_{r=1}^{k}\textbf{\textit{Prob}}_{U_{i_r}}\left(A_{i_r}\text{ is }\boldsymbol{f}_{i_r}\middle|o_{i_r}\right)\text{ where }B\equiv A_{i_1}\text{ is }\boldsymbol{f}_{i_1}\wedge\cdots\wedge A_{i_k}\text{ is }\boldsymbol{f}_{i_k}$$

In this way we find a support for each class and classify the example as having the class with highest support.

Notice that if each attribute universe Ω_j is partitioned using m_j fuzzy sets then there is an upper bound of $\prod_{j=1}^{n-1}m_j$ branches to any decision tree suggesting that the above calculation could be extremely computationally expensive. This is partially avoided, however, since because only triangular fuzzy sets are used a value has non zero membership only in two adjacent fuzzy sets. This means that for any attribute tuple $o=\left\langle o_1,\cdots,o_n\right\rangle$, $\textbf{\textit{Prob}}\left(B|o\right)$ is non zero for at most 2^{n-1} branches B. Precisely which branches these are can easily be determined so that unnecessary calculation may be avoided.

In many cases where we have formed a fuzzy partition of the classification space a method is required for defuzzifying from fuzzy sets to precise values. This is especially important for function approximation problems. More, precisely then we need a method by which when given a knowledge base of the form

$$\boldsymbol{Prob}\left(\text{A is } f_{\text{i}}\right)=\alpha_{\text{i}} \text{ for } \text{i}=1,\cdots,\text{n}$$

we can infer a value for A where it is supposed here that the universe of A is some interval of the real numbers.
Now given a fuzzy restriction of the form $\left(\text{A is } f_{\text{i}}\right)$ a standard defuzzification procedure is to take the average, assuming a uniform prior, of the values with membership 1 in f_{i}. We adopt this method here (see [3]) to obtain a set of n defuzzified values each with associated probability α_{i}. A single defuzzification value is then obtained simply by taking the expected value of these relative to the given probability distribution. In other words, if $\left(\text{A is } f_{\text{i}}\right)$ is defuzzified to v_{i} the final output value is given by $\text{v}=\sum_{\text{i}=1}^{\text{n}}\alpha_{\text{i}}\text{v}_{\text{i}}$

5 Fril Extended Rule Representations of Fuzzy Decision Trees

In some contexts it is desirable to have rule representations of Decision trees. For classical discrete decision trees first order logic conditionals will suffice but for probabilistic classifiers clearly these are inappropriate. The extended Fril rule provides an ideal way of representing decision trees with associated probability values within the unified uncertainty framework of Fril. The syntax of the extended rule is as follows;

$$(\text{h if } (\text{b}_1,\cdots,\text{b}_\text{n})):\left(\left(\text{u}_1,\text{v}_1\right)\cdot\cdot\left(\text{u}_\text{n},\text{v}_\text{n}\right)\right)$$

where h represents a head of the form (<pred> arguments) and b_{i} represents a list or conjunction of goals $(\text{c}_1,\cdots,\text{c}_\text{m})$ where c_i is of the form (<pred> arguments). In addition, $\left[\text{u}_\text{i},\text{v}_\text{i}\right]$ is an interval containing $\boldsymbol{Prob}\left(\text{h}|\text{b}_\text{i}\right)$ where the list of goals b_i is interpreted as a disjunction of goals. In the case where b_i corresponds to a crisp event then it is assumed that the set of events $\left\{\text{b}_\text{i}|\text{i}=1,\cdots,\text{n}\right\}$ are mutually exclusive and exhaustive. If on the other hand b_i corresponds to a fuzzy event of the form $\wedge_{\text{j}=1}^{\text{m}}\left(\text{A}_\text{j} \text{ is } f_{\text{i,j}}\right)$ for $\text{i}=1,\cdots,\text{n}$ then it is required that $\bigcup_{\text{i}=1}^{\text{n}}\times_{\text{j}=1}^{\text{m}}f_{\text{i,j}}$ is a fuzzy partition of $\times_{\text{j}=1}^{\text{m}}\Omega_\text{j}$ where Ω_j is the universe of A_j and the fuzzy cross product is defined using the product conjunction. Given supports for the body terms b_i for $\text{i}=1,\cdots,\text{n}$ the support for h is evaluated using an interval version of Jeffrey's rule corresponding to Jeffrey's rule in the case of point supports. (See [3] for details
)
. Clearly then by the properties of fuzzy partitions we may represent a fuzzy probabilistic decision tree as a set of extended rules where each rule corresponds to a particular classification. For instance, the decision tree from example 4.1 can be represented by the rule;

$$\begin{aligned}&((\text{Outcome is } \boldsymbol{success}) \text{ if } (\\&\quad ((\text{Score is } \boldsymbol{high\_score}) \text{ and } (\text{Players is } \boldsymbol{few}))\\&\text{or } ((\text{Score is } \boldsymbol{high\_score}) \text{ and } (\text{Players is } \boldsymbol{many}))\\&\text{or } ((\text{Score is } \overline{\boldsymbol{high\_score}}) \text{ and } (\text{Players is } \boldsymbol{few}))\\&\text{or } ((\text{Score is } \overline{\boldsymbol{high\_score}}) \text{ and } (\text{Players is } \boldsymbol{many}))\\&\quad)):((0.8297\ \ 0.8297)(0.5033\ \ 0.5033)(0.2309\ \ 0.2309)(0.1542\ \ 0.1542))\end{aligned}$$

6 The Application of Fuzzy Probabilistic Decision Trees to Function Approximation and Classification Problems

We shall now discuss the performance of the above fuzzy ID3 algorithm with respect to four test problems. The first three of these are model problems of a strongly non linear nature and the third is a real world problem from the field of vision.

Example 6.1

Consider the problem of classifying points in $[-1.5, 1.5]^2$ as legal if they lie within the ellipse $y^2 + 2x^2 = 1$ and illegal otherwise given a database of triples $\langle \text{CLASS}, \text{X}, \text{Y} \rangle$.

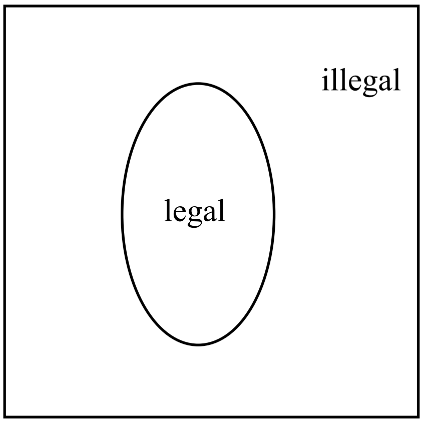
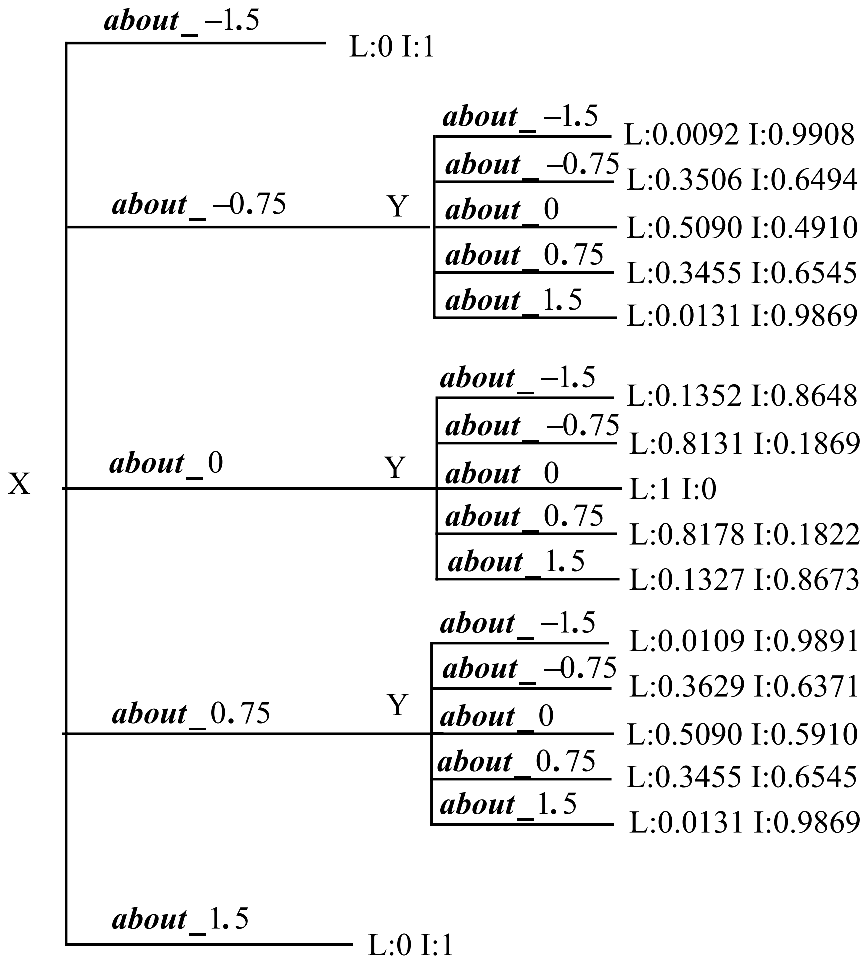


Figure 2

Here the database D consists of 126 triples generated by selecting random points from $\boxed{[-1.5, 1.5]^2}$ and labelling them with their classification value.
The X and Y universes are partitioned into 5 evenly spaced triangular fuzzy sets ;

***about* _-1.5** =[-1.5:1 -0.75:0]
***about_* -0.75** =[-1.5:0 -0.75:1 0:0]
***about_* 0** = [-0.75:0 0:1 0.75:0]
***about_* 0.75** = [0:0 0.75:1 1.5:0]
***about_* 1.5** =[0.75:0 1.5:1]

Using the fuzzy ID3 algorithm we obtain the following decision tree



Note that since this is a binary problem only one rule is given and the probabilities for illegal can be calculated trivially.

These decision rules correctly classified 100% of the training data set D and 99.168% of a test database consisting of 960 points forming a regular grid on $\boxed{[-1.5,1.5]^2}$. The decision surface for the positive quadrant is given in figure 4 below.

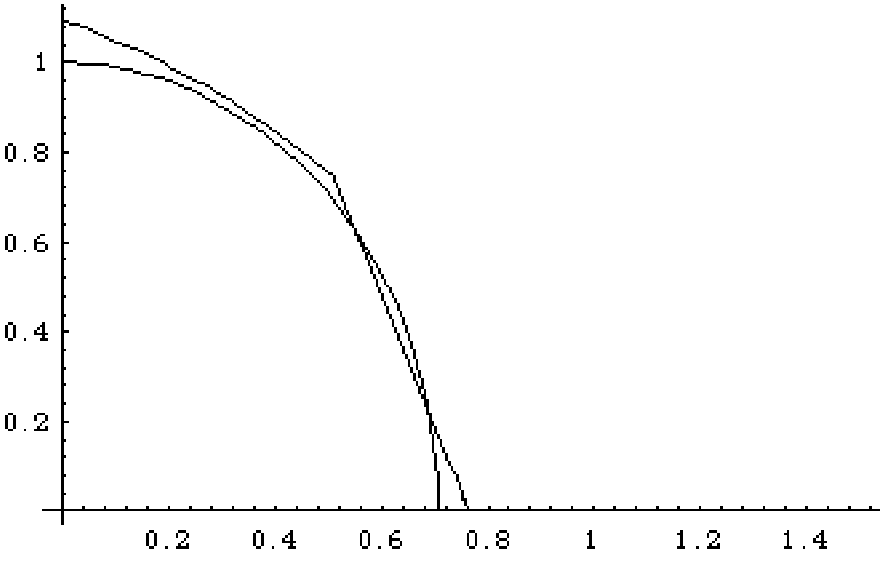
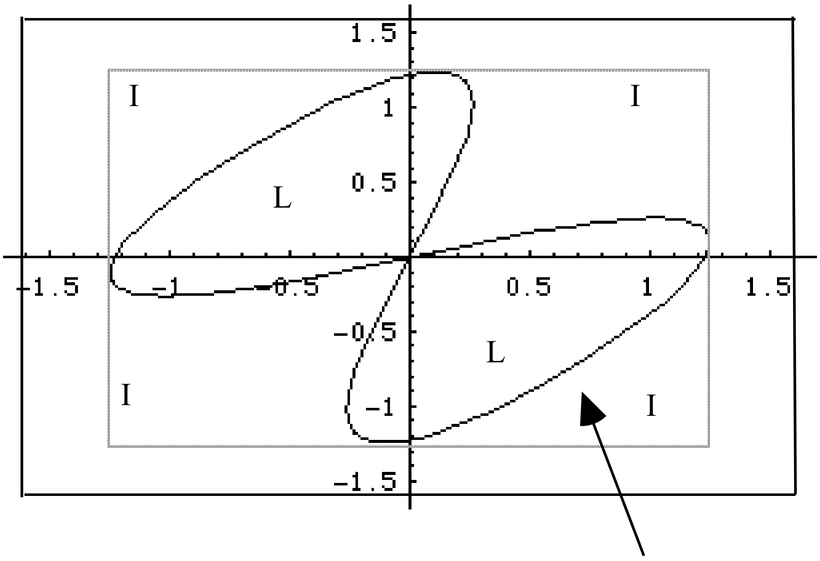


Figure 3

As mentioned before the incorporation of fuzzy sets into decision rules facilitates their use in function approximation problems. The following two examples demonstrate their potential in this area.

Example 6.2
In this problem a figure eight shape was generated according to the parametric equation $x = 2^{-0.5}(\boldsymbol{sin}\,2t - \boldsymbol{sin}\,t)$, $y = 2^{-0.5}(\boldsymbol{sin}\,2t + \boldsymbol{sin}\,t)$ where $t \in [0, 2\pi]$. Points in $[-1.5, 1.5]^2$ are classified as legal if they lie within the figure and illegal if they lie outside. The database consisted of a 960 points from a regular grid on $[-1.6, 1.6]^2$. Initially a legal / illegal intersection region was established by finding the intersection of the smallest two dimensional interval containing all the legal points and the smallest interval containing all the illegal points. In this case the intersection region contains all the legal points in the data base.



legal/illegal intersection region

Figure 4

All points outside the intersection region are therefore classified as illegal. For the intersection region the X and Y universes where evenly partitioned into 6 triangular fuzzy sets respectively and a fuzzy ID3 tree with 36 branches was generated on this region. The tree classified 95% of from a regular grid of test points correctly and the decision surface is given below.

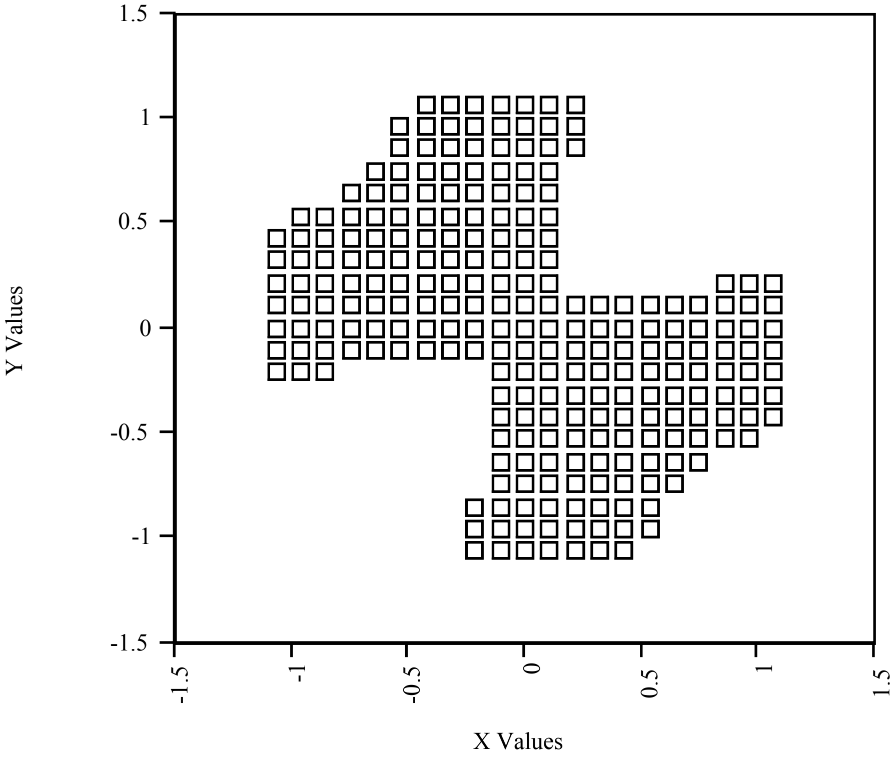


Figure 5

Example 6.3

In this example we consider a function approximation problem involving a complex continuous function. Here the database consists of 528 triples $\langle X, Y, \boldsymbol{sin}\, XY \rangle$ where the pairs $\langle X, Y \rangle$ form a regular grid on $[0,3]^2$. Due to the complexity of the function on this occasion 10 equally spaced triangular fuzzy sets are used to partition the independent variable domain $[0,3]$. These are;

***about*_ 0** = [0:1 0.333333:0]
***about*_0.3333** = [0:0 0.333333:1 0.666667:0]
***about*_ 0.6667** = [0.333333:0 0.666667:1 1:0]
***about*_ 1** = [0.666667:0 1:1 1.33333:0]
***about*_ 1.333** = [1:0 1.33333:1 1.66667:0]
***about*_1.667** = [1.33333:0 1.66667:1 2:0]
***about*_ 2** = [1.66667:0 2:1 2.33333:0]
***about*_2.333** = [2:0 2.33333:1 2.66667:0]
***about*_ 2.6667** = [2.33333:0 2.66667:1 3:0]
***about*_ 3** = [2.66667:0 3:1]

As in the previous example the dependent variable domain [-1, 1] is partitioned according to the algorithm described in section 2 into 5 fuzzy classes;

***class*_ 1** = [-1:1 0:0]
***class* _2** = [-1:0 0:1 0.380647:0]

***class_* 3** = [0:0 0.380647:1 0.822602:0]
***class_*4** = [0.380647:0 0.822602:1 1:0]
***class_*5** = [0.822602:0 1:1]

The fuzzy ID3 algorithm is used to generate a decision tree with 100 branches. The percentage error on a regular test database of 1024 points was 4.22427% and the decision surface together with true values is given below in figure 6 .

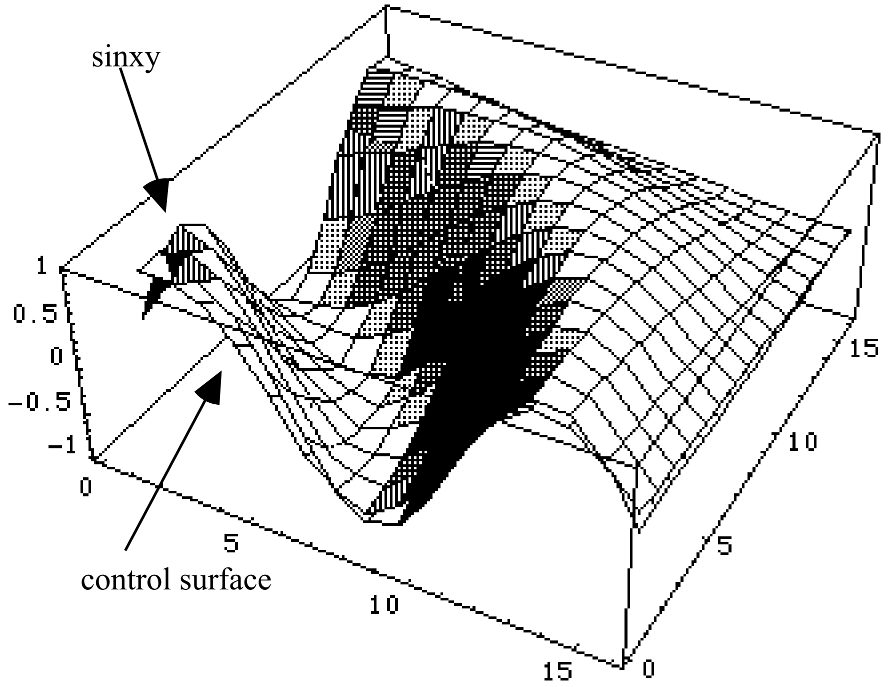
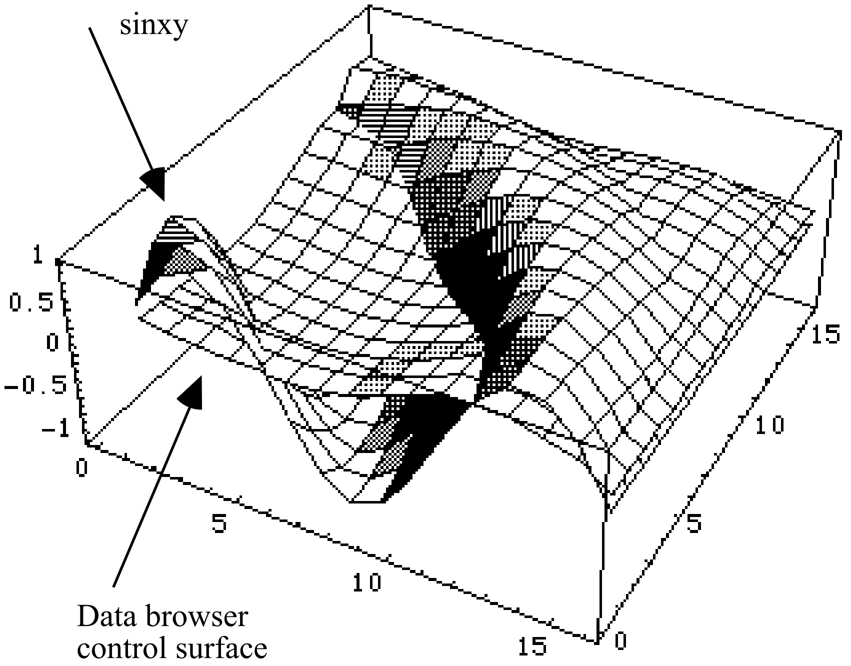


Figure 6

This result compares favorably with many other fuzzy engineering approaches applied to the problem . For example, a direct application of the Fril data browser (See [5]) to form Fril conditional rules leads to considerable decomposition errors as can be seen form the control surface shown below. It should be noted, however, that for problems of such complexity the Data browser gives much better results if used in conjunction with a clustering algorithm such as Kohonen [14]



Example 6. 4

The following example is motivated by a project to construct a system for the automatic classification of outdoor scenes (see [11]) given a set of eight measured features. The database consists of 3751 vectors corresponding to the feature values from distinct segments of about 200 images together with their classification class. Each segment of an image is to be classified as one of the following 11 classes.

1. “Cloud / Mist”
2. “Vegetation”
3. “Road Marking”
4. “Road Surface”
5. “Road Border”
6. “Building”
7. “Bounding Object”
8. “Road Sign”
9. “Signs / Poles”
10. “Shadow”
11. “Mobile Objects”

Classification is to be based on one of the following 8 features all of which are scaled so that their value lies in the interval [0, 1].

A1 Intensity
A2 Red - Green
A3 Yellow - Blue
A4 Size
A5 X co-ordinate
A6 Y co-ordinate
A7 Vertical orientation

A8 Horizontal orientation

Initially each of the attribute universes was partitioned into 7 fuzzy sets and the tree generated to a maximum depth of 4 from the training set. The entropy criterion selected the attributes Intensity, Red-Green, Yellow-Blue, X co-ordinate, Y co-ordinate to appear in a decision tree with 291 branches. Testing on the training set the latter classified correctly 69.0482% of the training set and 67.2064% of a test set of 7535 points. These results compare favourably with direct neural network and fuzzy cross product methods (see [10]).

7 Conclusions

The extension of ID3 to allow fuzzy sets as attribute values and classification classes has been shown to resolve many of the traditional difficulties associated with applying decision tree methods to real word problems. In particular, this approach allows for a more natural and robust treatment of continuous valued attributes. Furthermore, the use of fuzzy classification values together with a suitable defuzzification procedure means that fuzzy ID3 can successfully be applied to function approximation problems.

8 References

[1] J.F. Baldwin, T.P. Martin, B.W. Pilsworth, “FRIL Manual (Version 4.0)”, FRIL Systems Ltd, Bristol Business Centre, Maggs House, Queens Road, Bristol BS8 1QX, UK, 1988.

[2] J.F. Baldwin, “Computational Models of Uncertainty Reasoning in Expert Systems”, Computers Math. Applic. Vol. 19, No. 11 pp105-119, 1990.

[3] J.F. Baldwin, T.P. Martin, B.W. Pilsworth, “FRIL -Fuzzy and Evidential Reasoning in A.I”, Research Studies Press, John Wiley, 1995.

[4] J.F. Baldwin, J. Lawry, T.P. Martin, “A Mass Assignment Theory of the Probability of Fuzzy Events”, Fuzzy Sets and Systems, Vol. 83 pp353-367, 1996.

[5] J.F. Baldwin, T.P. Martin, “A Fuzzy Data Browser in Fril”, Fuzzy Logic (Ed J.F. Baldwin), John Wiley & Sons Ltd, 1996.

[6] J.F.Baldwin, J.Lawry, T.P.Martin, “A Note on the Conditional Probability of Fuzzy Subsets of a Continuous Domain”, to appear in Fuzzy Sets and Systems, 1996

[7] J.F.Baldwin, J.Lawry, T.P.Martin, “A Note on Probability / Possibility Consistency for Fuzzy Events”, Proceedings of IPMU 96 Vol. 1 pp521-526, 1996.

[8] J.F. Baldwin, J.Lawry, T.P. Martin, “A Mass Assignment Theory Approach to Fuzzy Rule Generation”, ITRC report, 1996.

[9] J.F. Baldwin, J.Lawry, T.P. Martin, “A Mass Assignment Based ID3 Algorithm for Decision Tree Induction”, to appear in the International Journal of Intelligent Systems (1997)

[10] J.F. Baldwin, T.P. Martin, J.G. Shanahan, “Fuzzy Logic Methods in Vision Recognition”, Fuzzy Logic: Applications and Future Directions, 1997

[11] N.W. Campbell, W.P.J. Mackeown, B.T. Thomas, T. Troscianko, “Interpreting Image Databases by Region Classification”, to appear in Pattern Recognition, 1997

[12] U.Fayyad, K.B.Irani, “On the Handling of Continuous-Valued Attributes in Decision Tree Generation”, Machine Learning 8, 87-102, 1992.

[13] R.C.Jeffrey, “The Logic of Decision”, Gordon & Breach Inc., New York, 1965.

[14] T.Kohonen, “Self-organizing Formation of Topologically Correct Feature Maps”, Biological Cybernetics 43 Vol. 1, pp59-69, 1982.

[15] J.R.Quinlan, “Induction of Decision Trees”, Machine Learning 1 pp81-106, 1986.

[16] J.R.Quinlan, “Decision Trees as Probabilistic Classifiers”, Proceedings of the fourth International Workshop on Machine Learning, 1987.

[17] J.R.Quinlan, “C4.5: Programs for Machine Learning”, San Mateo: Morgan Kaufmann, 1993.

[18] E.H.Ruspini, “A New Approach to Clustering”, Information and Control 15 pp22-32, 1969.

[19] M.Umano, H.Okamoto, I.Hatono, H.Tamura, F.Kawachi, S.Umedzu, J.Kinoshita, “Fuzzy Decision Trees by a Fuzzy ID3 Algorithm and its Application to Diagnosis Systems”, Proceedings of the third IEEE International Conference on Fuzzy Systems pp2113-2118, 1994.

[20] L.A. Zadeh, “Probability Measures of Fuzzy Events”, Journal of Mathematical Analysis and Applications 23, pp421-427, 1968.

[21] J.Zeidler, M.Schlosser, “Continuous - Valued Attributes in Fuzzy Decision Trees”, Proceedings IPMU 96 Vol. 1 pp395-400, 1996.