Inducing Fuzzy Decision Trees in Non-Deterministic Domains using CHAID

Jay Fowdar, Zuhair Bandar, Keeley Crockett

The Intelligent Systems Group
Department of Computing and Mathematics
John Dalton Building
Manchester Metropolitan University
Chester Street
Manchester
M1 5GD
United Kingdom
J.Fowdar@mmu.ac.uk, Z.Bandar@mmu.ac.uk, K.Crockett@mmu.ac.uk

Abstract

Most decision tree induction methods used for extracting knowledge in classification problems are unable to deal with uncertainties embedded within the data, associated with human thinking and perception. This paper describes the development of a novel tree induction algorithm which improves the classification accuracy of decision tree induction in non-deterministic domains. The research involved applies the principles of fuzzy theory to the CHAID (Chi-Square Automatic Interaction Detection) algorithm in order to soften the sharp decision boundaries which are inherent in traditional decision tree algorithms.
CHAID is a decision tree induction algorithm with the main feature of significance testing at each level, leading to the production of trees which require no pruning.
The application of fuzzy logic to CHAID decision trees can represent classification knowledge more naturally and in-line with human thinking and are more robust when it comes to handling imprecise, missing or conflicting information. The results of applying fuzzy logic to CHAID induced decision trees are presented in this paper. These have been obtained from sets of real world data, and show that the new fuzzy inference algorithm improves the accuracy over crisp CHAID trees. The results show that the increase in performance is dependant upon the inference technique employed and the amount of fuzzification applied.

Introduction

Knowledge acquisition today represents a major knowledge engineering bottleneck. (Michalski 1986) Attempts have been made to solve this problem by using computer programs to extract knowledge. One popular method for decision making or classification is systems inducing symbolic decision trees where rules can be extracted from the tree and used in a rule-based decision system. One such methodology which popularized the use of decision trees is the ID3 algorithm (Quinlan 1985). The decision tree induction process consists of two major components. Firstly a procedure to build the symbolic tree, and secondly an inference process for the actual decision making. The decision trees explored within this paper are constructed using the CHAID (Kass 1979) algorithm. CHAID is a learning algorithm that constructs a set of induction rules, which are capable of classifying objects, by analysing a training set where the classification of objects have been previously established. The original algorithm, proposed by Kass, is an offshoot of the Automatic Interaction Detection (AID) (Morgan 1963) technique designed for a categorized dependant variable. Important modifications from AID to CHAID include built-in significance testing, resulting in the most significant attribute is chosen for splitting in contrary to the most explanatory, and the formation of multi-way splits as apposed to binary splits.
The CHAID technique distinguishes itself from other decision tree induction techniques by its unique dynamic branching strategy. This is to find the optimal number of branches by grouping attribute values that are not significantly different in a single group. This branching strategy provides the algorithm with an in-built pruning mechanism for building decision tree models in non-deterministic domains, allowing it to effectively handle 'noisy' data.
The most apparent weakness of the CHAID algorithm is identified as the sharp decision boundaries that are created when an attribute is selected for splitting. Such a strict decision threshold can result in cases being predicted incorrectly due to a number of reasons, such as human error or measurement inaccuracies. The purpose of the research described in this paper is to use fuzzy logic to relax these boundaries thus improving the overall performance of CHAID induced trees. The application of fuzzy logic to the CHAID algorithm is intended to develop a more natural language approach to decision making.

The CHAID Algorithm

The CHAID algorithm is a highly efficient statistical technique for segmentation, or tree growing. Using as a

criterion the significance of a statistical test, CHAID evaluates all of the values of a potential predictor variable. The statistical test used is the Chi-Square test, which reflects how similar or associated variables are.
The algorithm merges values that are judged to be statistically homogeneous (similar) with respect to the target variable and maintains all other values that are heterogeneous (dissimilar). The algorithm then goes on to select the best predictor variable to form the first branch in the decision tree, such that each node is made of a group of similar values of the selected variable. The process continues recursively until the tree growth is complete.
A description of how a CHAID induced decision tree is now described:

1. For each predictor variable, X, find the pair of categories of X that is least significantly different (the greatest p-value), with respect to the target variable, Y. The method used to calculate this p-value depends upon the measurement level of Y.
    - If Y is continuous, use an F test.
    - If Y is nominal, form a two-way cross tabulation with categories of X as rows, and categories of Y as columns. Use the Pearson chi-squared test or the likelihood ratio test.

2. For the pair of categories of X with the largest p-value, compare the p-value to a pre-specified alpha level, α_{merge}.
    - If the p-value is greater than α_{merge}, merge this pair into a single compound category. As a result a new set of categories of X are formed, and the algorithm re-examines this predictor, proceeding from step 1 again.
    - If the p-value is less than α_{merge}, proceed on to step 3.

3. Calculate the adjusted p-value using a proper Bonferroni multiplier (Hommel 1999).

4. Select the predictor variable X that has the smallest adjusted p-value, i.e. the one which is most significant. Compare this value to a pre-specified alpha split level, α_{split}.
    - If the p-value is less than or equal to α_{split}, split the node based upon the set of categories of X.
    - If the p-value is greater than α_{split}, then this is a terminal node, do not split.

5. Continue the tree growing process until all stopping rules have been met.

Mentioned in steps 1 and 2, is the use of a p-value, which is obtained from the Chi-Squared test of significance. This test gives a numerical representation of how similar or associated the categories are, and is calculated from the contingency tables using equation 1.

$$\text{Chi Squared value } \chi^2 = \sum_{\text{all cells}} \frac{(\text{Observed Cell Count} - \text{Expected Cell Count})^2}{\text{Expected Cell Count}} \quad (1)$$

The use of an F-Test is referred to in Step 1. This test employs the F statistic, a ratio of two squares, to test various statistical hypotheses about the mean (or means) of the distributions from which a sample or a set of samples have been drawn.
Step 3 adjusts the p-values by applying the Bonferroni multiplier. This in effect is a process to determine the number of ways a predictor of any given type can be reduced to its most significant contingency table. This is then used in the Bonferroni inequality to obtain a bound for the significance level. The formula for calculating these multipliers, where a category predictor containing *c* categories is reduced to *r* groups, where ($1< r \leq c$), can be derived from the binomial coefficient as described in equations 2 and 3.

For an ordinal category $B_{Ordinal} = \binom{c-1}{r-1}$ (2)

For a nominal category $B_{\text{Nominal}} = \sum_{i=0}^{r-1} (-1)^i \frac{(r-i)^c}{i!(r-i)!}$. (3)

Step 5 describes the tree being grown until the stopping conditions are met. Tree growth will cease if any of the following conditions are met:

- All cases in a node have identical values for all predictors.
- The node becomes *pure*. This means all cases in the node have the same target value.
- If a depth has been specified, tree growth will stop when the depth of the tree has reached its limit.

For experimentation, the values of α_{split} and α_{merge} were both set at 0.05 giving a 95% significance level of the splits and the merged categories.

Fuzzy Sets and Fuzzy Logic

The concept of fuzzy sets has originated from Zadeh's pioneering paper (Zadeh 1965), where he stated that probabilities were an insufficient form of representation for uncertainty in Artificial Intelligence. By allowing certain amounts of imprecision to exist, Fuzzy Logic has played an important role in the management of uncertainty, especially in the field of Expert Systems.
The transition of object classification in a crisp set with defined boundaries is abrupt, in comparison to a fuzzy set, where the transition of the classes between objects is gradual.

Membership Functions

A membership function is essentially a curve that defines how each point in an input space is mapped to a membership value between 0 and 1. There are many different types of fuzzy membership functions which have been suggested and used in applying fuzzy logic (Kulkarni 2001). There is no definitive method suggested for the selection of a membership function, but there are experiments (Pedrycz 1998) which exist to help determine which is best to be used. The selection of the experiment depends heavily upon the specifics of the application, in particular, the way in which the uncertainty is manifested and captured during the experiment.

For the purpose of this paper linear membership functions have been used, however it should be noted that non-linear membership functions can also be used.

Equation 4 describes a linear increasing membership function.

$$f(x;a,b) = \begin{cases} 0 & \text{for } x < a \\ \dfrac{x-a}{b-a} & \text{for } a \le x < b \\ 1 & \text{for } b \le x \end{cases} \qquad (4)$$

where

a is the lower bound of f, also generating the zero membership value

b is the upper bound of f, also generating the maximum membership value

x is the value being evaluated

Inference Techniques

Knowledge from decision trees is often represented as rules, which consist of two primary parts, an antecedent and a consequence. Fuzzy rules are commonly described as:

IF [V_1 is v_1] ^ [V_2 is v_2] THEN [V_c is v_c].

The rules in a fuzzy relation are combined and an output can then be inferred. Inference can be used as a tool for reasoning to deduce an outcome from a set of facts. The inference technique combines the information obtained from firing a number of IF…THEN rules, and can be seen to consist of four main parts:

- Combining the data from the antecedent of a particular rule
- Applying the resultant value to the consequence of that particular rule
- Combining the resultants from all the rules
- Interpreting the outcome

For fuzzy inference, the inputs must first be fuzzified, which is the process of determining the fuzzy sets used to describe the values of the linguistic variables. The knowledge base must then be constructed, which is where the IF…THEN rules, consisting of fuzzy antecedents and consequents, are defined. A fuzzy value outcome is then produced by combining the information obtained from firing the rules. Finally a non-fuzzy value is produced by the defuzzification process that best describes the fuzzy value outcome.

Within classical set theory, there are single definitions for the two set operators, intersection and union. In fuzzy set theory, there are many different suggestions of alternative interpretations. Classical set theory restricts outcome values to 0 and 1, whereas if the membership grades are between these boundaries, the results are different.

Zadeh's Original Fuzzy Operators

In classical set theory, basic operations exist such as union, intersection and complement. These operations can also be defined for fuzzy sets too. As there could be an infinite number of membership values, infinitely many different interpretations can be assigned to these operations.

Zadeh proposed the operators, *min* and *max,* for intersection and union respectively as extensions from the corresponding actions upon crisp sets. These operators are the most commonly used and are special in that if the sets become restricted, the operators act in the same manner as those for crisp sets.

As a brief summary, the intersection and union operators proposed by Zadeh are presented in Table 1.

Weak and Strong Operators

Strengths and weaknesses can be assigned to operators by using parameters. The majority of operators are parameterised by using a weight w. The result of introducing this parameter has a significant effect on the outcome produced by the function. The significance of the selection of this parameter is discussed by Yager (Yager 1997), where he suggests a methodology that has been used by many researchers in this field.

Yager makes the min and max operators more adaptable by employing the use of a parameter, w, to soften the operator. A different fuzzy intersection or union is obtained by varying the parameter w, which lies in the range of $[0…\infty]$, it can therefore be implied that w determines the strength of the operation.

In natural language we generally use the AND conjunction to imply we strongly require more than one condition to be true. Yager's methodology implies that given a logical statement consisting of conditions and a resultant (IF…THEN statement), where the conditions implement the use of the AND operator, we regard the parameter, w as inversely proportional to the strength of the AND. It can therefore be assumed that w is a measure of how strongly we require the conditions to be true.

Yager's Intersection and Union operators can be found within Table 1, as well as other well recognised sets of theoretical operators for fuzzy union and intersection, which have been employed in the Fuzzy CHAID Induction Algorithm (Fuzzy-CIA).

The degree of the intersection operator is dependant on the value $1/w$. As $w \rightarrow \infty$ the lowest membership grade in each set dictates the degree of membership in their intersection.

Reference	Fuzzy Unions	Fuzzy Intersection	Range of Parameter
Zadeh	$\max[a, b]$	$\min[a, b]$	n/a
Hamacher	$\frac{a+b-(2-q)ab}{1-(1-q)ab}$	$\frac{ab}{q+(1-q)(a+b-ab)}$	$Q \in (0,\infty)$
Yager	$\min\left[1,\left(a^w+b^w\right)^{1/w}\right]$	$1-\min\left[1,\left((1-a)^w+(1-b)^w\right)^{1/w}\right]$	$w \in (0,\infty)$
Dubois & Prade	$\frac{a+b-ab-\min(a,b,1-\alpha)}{\max(1-a,1-b,\alpha)}$	$\frac{ab}{\max(a,b,\alpha)}$	$\alpha \in (0,1)$

Table 1. Fuzzy Inference Operators

Mamdani Inference

One type of Fuzzy Rule Based System (FRBS) was developed by Mamdani (Mamdani 1975), who was able to augment Zadeh's initial formulation in a way which allowed it to be applied to a fuzzy control system. These types of fuzzy systems are commonly known as fuzzy logic controllers. As before, fuzzification enables Mamdani-type FRBS's to handle crisp input values, mapping from crisp values to fuzzy sets defined in the universe of discourse of that input. The inference system establishes a mapping between the fuzzy sets in the input domain and the fuzzy sets in the output domain. The defuzzification interface transforms the fuzzy output from the fuzzy rule base into a non-fuzzy output. The defuzzification interface has to aggregate the information provided by the output fuzzy sets and to obtain a crisp output value from them. This can be done in two ways, Mode A-FATI (first aggregate, then infer) or Mode B-FITA (first infer, then aggregate). Recently the Mode B method has become more popular (Cordon, Herrera, Peregrin 1997), as real time applications require a faster response time.

For Mode B-FITA the contribution from each fuzzy set is considered separately and the final crisp value is obtained by means of averaging the set of crisp values derived from each of the fuzzy sets. The defuzzification interface aggregates the individual output fuzzy sets by means of the maximum fuzzy union:

$$\mu(y) = \max \{\mu B'_1(y), \mu B'_2(y), \mu B'_3(y), \mu B'_n(y)\}$$

The most common choice for this is the centre of gravity (CG) weighted by the matching degree, whose value is calculated by equation 5

$$y_0 = \frac{\sum_{i=1}^{m} h_i \cdot y_i}{\sum_{i=1}^{m} h_i} \qquad (5)$$

Where y_i is the CG inferred from rule i and $h_i = \mu_A(x_0)$ being the matching between the system input x_0 and the rule antecedent. This approach reduces the computational burden of aggregating the rule outputs to the fuzzy set B', as used in Mode A-FATI.

Fuzzy CHAID Induction Algorithm

This paper has introduced a novel Fuzzy-CIA. Firstly, a crisp CHAID tree must be induced, using the methodology described earlier, from a training set established from the cross-validation method (Stone 1978). Tree growth was continued until all stopping criteria were met to produce the optimal tree. The next stage is to introduce some fuzzification to these membership functions, allowing partial degrees of memberships in all branches throughout the tree. To achieve this, a small area around the split threshold point must be defined. One common statistical method of defining the spread of data is the Standard Deviation. For each numeric attribute used within the data set, the standard deviation can be calculated from the training set. Multiples of the standard deviation can be added or subtracted from the split value to derive a partition of the domain, which would have partial degrees of membership. The aim of the fuzzification is to correctly classify records which are in or around the threshold splitting value. For this reason small multiples of the standard deviation were experimented with, as excess amounts of fuzzification would generalize the tree too much, reducing classification accuracy. For a dataset where the outcome is discrete, an inference technique described in Table 1 is selected. The final classification of the record is determined by the leaf node which possesses the highest membership grade after union.

Applying Mamdani To CHAID

For the dataset where the outcome variable is numeric, the Mamdani inference technique is employed, using Mode-B FITA for defuzzification to obtain a final predicted value.

In order to evaluate the performance of Fuzzy-CIA in numeric datasets, the Boston Housing Dataset was utilised. For this dataset the Mamdani methodology has been selected as the fuzzification inference technique, with some modification. Typically the Mamdani inference technique is applied to datasets where the output variable is ordinal. For example, if we were to consider the quality of a product, the output fuzzy set would contain the members {*bad quality, medium quality, good quality}*. However to describe a continuous output it is necessary to first discretise the output into groups. Figure 1 shows the

membership function of the output attribute Median Value split into 3 groups. By increasing the number of groups it is possible to improve the performance of the aggregation phase of defuzzification.

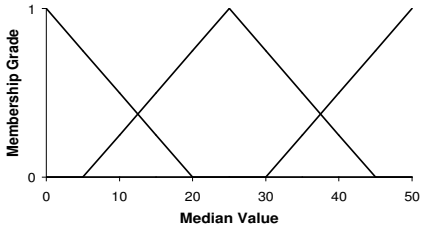

Figure 1–Membership function for output variable median value.

Evaluating Performance

Two different methods are employed for evaluating the performance of the Fuzzy-CIA tree, and the selection of method is dependant upon the type of the target variable. When calculating the percentage accuracies for a dataset where the target variable is discrete, the number of correct classifications of each outcome will be determined separately and an overall measure of the performance can be obtained from finding the average of these values. The performance of the tree can be obtained from equation 6.

$$\text{Performance of Tree} = \frac{\sum_{i=1}^{i=n} \frac{\text{Number of Correctly Classified Records in Class } i}{\text{Total Number of Records in Class } i}}{n} \quad (6)$$

Where i = Class i, n= Number of classes

For the housing dataset, where the dependant variable is numeric, the performance of the tree is evaluated by measuring the error between the predicted Median Value of a house and the actual value. Since this difference can be negative, the result is squared, and an average squared error is calculated for the complete data set as follows, with a lower error indicating the performance of a more efficient tree. The calculation to obtain the performance of the tree is described in equation 7.

$$\text{Performance of Tree} = \frac{\sum_{x=1}^{x=r} (\text{Predicted Value} - \text{Actual Value})^2}{r} \quad (7)$$

Where r = Number of records in the dataset

The results presented in this paper use cross validation to obtain the average efficiency of the decision trees.

Data Sets

The Fuzzy CIA has been applied to four real world data sets. These are Bankloan (Attar 2003), Diabetes, Boston Housing and the Vehicle dataset (Blake & Merz 1998). The Bankloan data set decides whether to accept or reject an application for a loan. The Diabetes data set is concerned whether a patient shows signs of diabetes. The Vehicle data set is concerned with classifying a given silhouette as one of three type of vehicle. Finally, the Boston Housing dataset is concerned with the prediction of house value, and the outcome classification is numeric.

Results

Tables 2,3,4 and 5 show the average results obtained from applying fuzzification to CHAID induced decision trees for the data sets explored. The tables show the performance of the crisp CHAID trees compared with the different inference techniques which have been investigated. The first row in each table indicates the performance of a traditional crisp CHAID tree. The percentage shown is the average classification accuracy and also stated is the amount of fuzzification applied to the decision trees and the parameter used for inference.

Inference Technique	Highest Performance	Amount Of Fuzzification	Parameter
Crisp	69.52%	N/A	N/A
Yager	76.02%	2 S.D	w=10
Hammacher	72.11%	0.5 S.D	w=1
Dubois/Prade	74.49%	1.5 S.D	W=0.5

Table 2 – Bankloan Dataset

Inference Technique	Highest Performance	Amount Of Fuzzification	Parameter
Crisp	68.81%	N/A	N/A
Zadeh	74.00%	5.0 S.D	N/A
Yager	76.58%	7.8 S.D	W=10
Hammacher	73.53%	2.0 S.D	w=0.001
Dubois/Prade	74.25%	5.2 S.D	W=0.5

Table 3 – Diabetes Dataset

Inference Technique	Highest Performance	Amount Of Fuzzification	Parameter
Crisp	69.52%	N/A	N/A
Zadeh	70.81%	2.5 S.D	N/A
Yager	76.32%	1.5 S.D	w=10
Hammacher	72.11%	0.5S.D	w=1
Dubois/Prade	74.49%	5.2 S.D	w=0.5

Table 4 – Vehicle Dataset

Lowest Squared Mean Error	Amount Of Fuzzification	Number of Discretised Regions
20.981	CRISP	N/A
21.072	0.4 SD	10
20.650	0.5 SD	20
20.105	0.4 SD	30
20.241	0.6 SD	40
20.163	0.6 SD	50

Table 5 – Housing Dataset using Mamdani Inference

Discussion

The results obtained from the Fuzzy-CIA show a significant improvement in performance in comparison to the corresponding crisp CHAID tree. For the datasets where a discrete target variable is used, the improvement can be seen over all the different inference techniques which were used. The Zadeh min-max algorithm has a weakness in that there is no interaction between different variables, and the improvement achieved for this method is a result of the sharp decision boundaries being described as a series of fuzzy regions. For the other inference techniques, classification accuracy has been achieved by carefully selecting Intersection and Union parameters, with the Yager inference method showing the highest increase in classification accuracy. For the Bankloan dataset an improvement of 6.5% was achieved, the Diabetes dataset improved by 7.77% and the increase in performance from the Vehicle dataset was 6.8%.

The Housing dataset, where a numeric outcome is predicted, also showed improvement over the crisp tree. It was further shown that varying the number of output fuzzy sets affected the performance of the tree. The best performance was achieved when 30 fuzzy output sets were used, reducing the error rate by 0.876. The increase in the number of output fuzzy sets has allowed for improvement in the aggregation during the defuzzification process, as it is possible to include more fuzzy regions which are closer to the actual output variable, and not take into consideration other regions which are spread throughout the output domain.

Conclusion

A novel Fuzzy CHAID Induction Algorithm has been introduced in this paper which has improved the performance of decision trees induced using the CHAID algorithm. This has been achieved by introducing fuzzy logic to soften the sharp decision boundaries which are apparent in traditional crisp decision trees.

Five different inference techniques are explored and applied to three different real world datasets with discrete outcomes. A modified approach to the Mamdani inference methodology has allowed for fuzzy logic to be applied to the Housing dataset, where the output variable is numeric. Each of the different inference techniques have all showed significant improvements with respect to their relative crisp trees, in terms of their average classification accuracy.

References

Al-Attar, H 1996. “Improving The Performance Of Decision Tree Induction In Non-Deterministic Domains.” MPhil Thesis, Manchester Metropolitan University,

Cordon, O., F. Herrera, et al. 1997. "A Study on the Use of Implication Operators Extending the Boolean Conjunction in Fuzzy Control." 7th Int. Fuzzy Systems Association World Congress **3**: 243-248.

Hommel, G. and G. Bernhard 1999. "Bonferroni procedures for logically related hypotheses." Journal of Statistical Planning and Inference **82**(1-2): 119-128.

Kass, G. V. 1979. "An Exploratory Technique for Investigating Large Quantities of Categorical Data." Applied Statistics **29**(2): 119-127.

Kulkarni, A. D. 2001. Computer vision and fuzzy-neural systems. Upper Saddle River, NJ, Prentice Hall**:** xiii, 509.

Mamdani, E. H. and A. S. 1999. "An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller." Int. Journal of Human Computer Studies **51**: 135-147.

Michalski, R. S. 1986. "Understanding the Nature of Learning." Machine Learning: An Artificial Intelligence Approach **2**: 3-26.

Morgan, J. N. and J. Sonquist 1963. "Problems in The Analysis Of Survey Data, and A Proposal." American Statistical Association Journal **58**: 415-434.

Pedrycz, W. and F. Gomide 1998. An introduction to fuzzy sets : analysis and design. Cambridge, Mass., MIT Press.

Quinlan, J. R. 1985. "Induction Of Decision Trees." Machine Learning **1**: 81-106.

Stone, M. 1978. "Cross Validation : A review." Mathematische Operationsforschung Statistischen **9**: 127-139.

Yager, R. R. 1997. "On a class of weak triangular norm operators." Information Sciences **96**(1-2): 47-78.

Zadeh, L. A. 1965. "Fuzzy Sets." Information and Control **8**: 338-353.