Integration of Fuzzy Classifiers with Decision Trees

I-Jen Chiang
chiang@robot.csie.ntu.edu.tw

Jane Yung-jen Hsu
yjhsu@csie.ntu.edu.tw

Department of Comuter Science and Information Engineering
National Taiwan University
Taipei, Taiwan 107, R.O.C.

Abstract

It is often difficult to make accurate predictions given uncertain and noisy data for classification. Unfortunately, most real-world problems have to deal with such imperfect data. This paper presents a new model for *fuzzy classification* by integrating fuzzy classifiers with decision trees. In this approach, a *fuzzy classification tree* is constructed from the training data set. Instead of defining a specific class for a given instance, the proposed fuzzy classification scheme computes its degree of *possibility* for each class. The performance of the system is evaluated by empirically compared with a standard decision tree classifier C4.5 on several benchmark data sets the UCI machine learning repository.

1 Introduction

Classification techniques such as decision trees [4] have been widely used for discovering regularities in complex data. Successful applications include process control, pattern recognition, and diagnosis. Most existing classification techniques have difficulty in dealing with uncertain and noisy data [5,6,9]. Unfortunately, for many real world problems, uncertainty and noise in data cannot be ignored.

Previously, we have introduced the concept of *fuzzy classification trees* (FCT) [2] for domains with vague classifications. Rather than performing a two-stage process that couples decision trees with either *pre-fuzzification* [8,9] or *post-fuzzification* [1,3,7], the fuzzy classification tree presents a theoretically sound integration of fuzzy classifiers with decision trees. The structure is very robust with respect to a large amount of noise in the data for classification.

In this paper, we will present the algorithm for constructing fuzzy classification trees, as well as some empirical results on five different data sets from the UCI repository. Section 2 briefly reviews the definitions of fuzzy classification trees. Section 3 describes the basic algorithm for constructing a fuzzy classification tree from a data set. The empirical results comparing FCT with C4.5 [4] are summarized in section 4, followed by the conclusion.

2 Definitions

This section introduces the problem of *fuzzy classification*.

For any classification problem, the collection of all possible instances constitute the *instance space*, which is denoted by $\mathcal{X}$. Let $\mathcal{C} = \{C_1, C_2, \ldots C_n\}$ is a set of classes. A *classifier* or *decision function*, D, is a function that maps each instance into a class. That is, $D(\mathbf{x}) = C_i$, where $\mathbf{x} \in \mathcal{X}$ and $C_i \in \mathcal{C}$

A *fuzzy classifier* is a function, $\mathbf{F} : \mathcal{X} \longrightarrow \{\langle \wp_1, \ldots, \wp_n \rangle | \wp_i \in [0,1]\}$, such that each $\wp_i$ is a function defining the *possibility* that an instance belongs to the class C_i.

A *fuzzy classification tree* (FCT) is used to implement the fuzzy classifier. Given an FCT, let N_L denote the node labelled by L, and B_L denote the branch leading into node N_L. The children of N_L are labelled as $N_{L.i}$, where $i \in \{1, 2, 3, \ldots\}$. Each node N_L is associated with a class C_L and the possibility function P_L. Each branch B_L is associated with a membership function μ_L which is a function that defines the degree of possibility for any instance $\mathbf{x} \in \mathcal{X}$ to be classified as class C_L based on attribute A.. The possibility function P_L is the composition of the membership functions along the branches from root to node N_L. If N_L is the root node, P_L is set to be 1.

In general, there are many FCTs that implement the same fuzzy classifier. The entropy has been used to evaluate the FCTs[2]. The entropy function of node N_L is

$$\text{Info}(S_L) = -\sum_{\forall c \in \mathcal{C}} \frac{\mathcal{P}_L^c}{\mathcal{P}_L} \times \ln \frac{\mathcal{P}_L^c}{\mathcal{P}_L}.$$

The *entropy*, i.e., information content, of $\mathcal{T}_L$ can be defined as

$$\text{Info}_T(S_L) = \sum_{i=1}^{b_L} \frac{\mathcal{P}_{L.i}}{\mathcal{P}_L} \times \text{Info}(S_{L.i}).$$

where b_L is number of branches from node N_L, $\mathcal{P}_L$ be the sum of possibility $P_L(\mathbf{x})$ for all $\mathbf{x}$ in node N_L, $\mathcal{P}_L^c$ be the sum of possibility $P_L(\mathbf{x})$ for all $\mathbf{x}$ in node N_L of class $c \in \mathcal{C}$. The information gain at node N_L is defined by

$$\text{Gain}(\text{Test}_L) = \text{Info}(S_L) - \text{Info}_T(S_L)$$

due to the test Test_L.

3 Construction of Fuzzy Classification Trees

In this section, an algorithm for fuzzy classification is presented. The main algorithm for constructing an FCT is shown in Figure 1.

Initially, the system is given a set of training instances, S_1 denoted by S_0 in it, which is S_0. Let $\mathcal{L}$ contain the set of labels corresponding to the unexpanded leaf nodes of the tree. Let S_L denote the set of instances that have bee assigned to node N_L. The algorithm starts by creating a root node N_1, adding its label to $\mathcal{L}$, and initializing S_1 to be S_0.

Algorithm *BUILD_FCT*

[Input] A set of real-valued training instances S_0.

[Output] An FCT.

1. $L \longleftarrow 1$
2. $\mathcal{L} \longleftarrow \{1\}$
3. $S_1 \longleftarrow S_0$
4. *Until* $\mathcal{L} = \phi$
5. $\quad L \longleftarrow \text{random}(\mathcal{L})$
6. $\quad \mathcal{L} \longleftarrow \mathcal{L} \setminus \{N_i\}$
7. $\quad \forall a_i, T_L \longleftarrow \text{spawn\_new\_tree}(N_L, a_i)$
8. $\quad \text{Best} \longleftarrow T_k \text{ s.t. } \text{Info}(T_k) = \max_j \text{Info}(T_j)$
9. $\quad Gain \longleftarrow \text{Info}(T_L) - \text{Info}(\text{Best})$
10. $\quad$ *if* $Gain > \epsilon$
11. $\qquad$ *Add* the labels of all leaf nodes of Best into $\mathcal{L}$
12. $\qquad$ *Assign* subsets of S_L into $S_{L.1}, \ldots, S_{L.k}$

Figure 1: The algorithm for constructing an FCT.

The procedure spawn_new_tree(N_L, a_i) defines an expansion from node N_L according to attribute a_i. The details of procedure is shown in Figure 2. All attributes of an instance are viewed as coordinates in an n-dimensional Euclidean space.

Consider any cluster of class C_i along the coordinate a_i resulted from step 3. We first calculate its center of gravity by standard geometric method.

Suppose that there are k branches generated by an attribute. The procedure for computing the entropy of any given FCT is defined by the algorithm in Figure 3.

Algorithm *SPAWN_NEW_TREE*

[Input] A leaf node N_L and an attribute a_i.

[Output] A subset expanded from N_L.

1. $\forall j$ *Project* instances in S_L of class C_j onto attribute a_i
2. *Smooth* the resulting histograms using k-median method
3. *Partition* each smoothed histogram into clusters
4. *Create* a new branch from N_L for each cluster.
5. *Define* the membership function for each branch

Figure 2: The spawn_new_tree algorithm.

Algorithm *EVALUATE_ENTROPY*

[Input] An FCT with root node N_L.

[Output] The entropy value of $\mathcal{T}_L$.

1. $i \longrightarrow 0$
2. $\forall C_j$
3. $\quad i \longleftarrow i+1$
4. $\quad P_L^{C_j} = \sum_{\forall \mathbf{x} \in S_L} u_j(\mathbf{x})$
5. $\quad \mathcal{P}_{L.i} = \sum_{\forall \mathbf{x} \in S_L} u_i(\mathbf{x})$
6. $P_L = -\sum_{C_j \text{ at } N_L} P_L^{C_j}$
7. $\text{Info}(S_L) = -\sum_{C_j \text{ at } N_L} \frac{P_L^{C_j}}{P_L} \ln \frac{P_L^{C_j}}{P_L}$
8. $\text{Info}_T(S_L) = -\sum_{l=1}^{k} \frac{\mathcal{P}_{L.l}}{\mathcal{P}_L} \times \text{Info}(S_{L.i})$

Figure 3: The algorithm for evaluating the entropy of $\mathcal{T}_L$.

4 Empirical Results

We have tested our algorithm on five data sets from the UCI repository (ftp://ftp.ics.uci.edu/machine-learning-databases).

Golf This is sample data set with 14 artificially generated instances. We will use this data set to compare the trees that generate by C4.5 and FCT. There are two classes: "Play" and "Don't Play". Each instance is constructed by four attributes, which are categorical or numerical, associated with their possible values[4].

Monks' Problem The three Monks' problems are a collection of three binary classification problems over a six-attribute discrete domain. The classes is either 0 or 1. 6 categorical value attributes, no missing value. There are noisy data in *monk1* and *monk2*.

Ionosphere This data set is a binary classification task. The radar data was collected by a system in Goose Bay, Labrador. This system consists of a phased array of 16 high-frequency antennas with a total transmitted power on the order of 6.4 kilowatts. The targets were free electrons in the ionosphere. "Good" radar returns are those showing evidence of some type of structure in the ionosphere. "Bad" returns are those that do not; their signals pass through the ionosphere. There are 351 instances with all 34 numerical attributes, no missing value.

The result of comparing the accuracy of FCT with C4.5 on these problems is shown in Table 3. Note the tree that we use to compare is without pruning.

Table 3: Average accuracy between C4.5 and FCT over the golf, monks, and ionosphere problems.

	Golf	Monk1	Monk2	Monk3	Ionosphere
C4.5	100%	76.6%	65.3%	92.6%	96.5%
FCT	100%	86.5%	73.4%	93.2%	96.2%

The clustering method determines the performance of FCT. As the result on the *ionosphere* problem, the accuracy of FCT is lower than the accuracy of C4.5. Since the values of all the attributes are distributed in $[-1, 1]$, the size of the clusters has an effect on the accuracy of FCT. To improve the clustering method is one of our further objectives.

5 Conclusion

This paper has presented an algorithm that integrates the fuzzy classifiers with decision trees. The algorithm attempts to expand the FCT while minimizing its entropy at each step.

We have compared FCT with C4.5 with the empirical results of five data sets in the above section. From the noise-free data (Golf) to the data with a great amount of noise (Monk2), the accuracy rate of FCT is better than C4.5. C4.5 classifies an instance into exactly one class. The instances with attribute values around class boundaries are forced to be classified into a single class, which may result in wrong predictions, especially in the noisy domains. Instead of making a rigid classification, it is sometimes necessary to identify more than one possible classifications for a given instance.

FCTs allow multiple predictions to be made, each of which is associated with a degree of possibility. In application domains that involve a large amount of data with uncertainty, such as medicine or business, fuzzy classification trees can serve as a useful tool for generating fuzzy rules or discovery knowledge in database.

1. Z. Chi and H. Yan, "ID3-Derived Fuzzy Rules and Optimized Defuzzification for Handwritten Numeral Recognition," *IEEE Trans. Fuzzy Systems*, Vol.4, No.1, 1996, 24-31.
2. I. -J. Chiang and J. Y. -j Hsu, "Fuzzy Classification Trees," *Proc. of Ninth Internat. Symposium on Artificial Intelligence*, Cancon, Mexico, 1996.
3. K. J. Cios and L. M. Sztandera, "Continuous ID3 algorithm with Fuzzy Entropy Measures," *Proc. of the Internat. Conference on Fuzzy Systems*, 1992, 469-476.
4. J. R. Quinlan, *C4.5 programs for machine learning*, Morgan Kaufmann, San Mateo, CA, 1993.
5. M. Sugeno and G. T. Kang, "Structure Identification of Fuzzy Model," *Fuzzy Sets and Systems*, Vol.28, 1988, 15-33.
6. T. Takagi and M. Sugeno, "Fuzzy Identification of Systems and its Applications to Modeling and Control," *IEEE Trans. System Man Cybernet.*, Vol.5, 1985, 116-132.
7. T. Tani, M. Sakoda and K. Tanaka, "Fuzzy Modeling by ID3 Algorithm and its Applications to Prediction of Heater Outlet Temperature," *Proc. of Second IEEE Internat. Conference on Fuzzy Systems*, 1992, 923-930.
8. R. Weber, "Automatic Knowledge Acquisition for Fuzzy Control Application," *Proc. of the Internat. Symposium on Fuzzy Systems*, 1992, 9-12.
9. Y. Yuan and M. J. Shaw, "Induction of fuzzy decision trees," *Fuzzy Sets and Systems, 69*, 1995, 125-139.