

Chapter 10

Fuzzy Decision Trees

10.1 Overview

There are two main types of uncertainty in supervised learning: statistical and cognitive. Statistical uncertainty deals with the random behavior of nature and all techniques described in previous chapters can handle the uncertainty that arises (or is assumed to arise) in the natural world from statistical variations or randomness. While these techniques may be appropriate for measuring the likelihood of a hypothesis, they say nothing about the meaning of the hypothesis.

Cognitive uncertainty, on the other hand, deals with human cognition. Cognitive uncertainty can be further divided into two sub-types: vagueness and ambiguity. Ambiguity arises in situations with two or more alternatives such that the choice between them is left unspecified. Vagueness arises when there is a difficulty in making a precise distinction in the world

Fuzzy set theory, first introduced by Zadeh in 1965, deals with cognitive uncertainty and seeks to overcome many of the problems found in classical set theory. For example, a major problem in the early days of control theory is that a small change in input results in a major change in output. This throws the whole control system into an unstable state. In addition there was also the problem that the representation of subjective knowledge was artificial and inaccurate.

Fuzzy set theory is an attempt to confront these difficulties and in this chapter we present some of its basic concepts. The main focus, however, is on those concepts used in the induction process when dealing with fuzzy decision trees. Since fuzzy set theory and fuzzy logic are much broader than the narrow perspective presented here, the interested reader is encouraged to read [Zimmermann (2005)].

10.2 Membership Function

In classical set theory, a certain element either belongs or does not belong to a set. Fuzzy set theory, on the other hand, permits the gradual assessment of the membership of elements in relation to a set.

Definition 10.1 Let U be a universe of discourse, representing a collection of objects denoted generically by u . A fuzzy set A in a universe of discourse U is characterized by a membership function μ_A which takes values in the interval $[0, 1]$. Where $\mu_A(u) = 0$ means that u is definitely not a member of A and $\mu_A(u) = 1$ means that u is definitely a member of A .

The above definition can be illustrated on a vague set, that we will label as *young*. In this case the set U is the set of people. To each person in U , we define the degree of membership to the fuzzy set *young*. The membership function answers the question: “To what degree is person u young?”. The easiest way to do this is with a membership function based on the person’s age. For example Figure 10.1 presents the following membership function:

$$\mu_{Young}(u) = \begin{cases} 0 & \text{age}(u) > 32 \\ 1 & \text{age}(u) < 16 \\ \frac{32 - \text{age}(u)}{16} & \text{otherwise} \end{cases} \quad (10.1)$$

Given this definition, John, who is 18 years old, has degree of youth of 0.875. Philip, 20 years old, has degree of youth of 0.75. Unlike probability theory, degrees of membership do not have to add up to 1 across all objects and therefore either many or few objects in the set may have high membership. However, an objects membership in a set (such as “young”) and the sets complement (“not young”) must still sum to 1.

The main difference between classical set theory and fuzzy set theory is that the latter admits to partial set membership. A classical or crisp set, then, is a fuzzy set that restricts its membership values to $\{0, 1\}$, the end-points of the unit interval. Membership functions can be used to represent a crisp set. For example, Figure 10.2 presents a crisp membership function defined as:

$$\mu_{CrispYoung}(u) = \begin{cases} 0 & \text{age}(u) > 22 \\ 1 & \text{age}(u) \leq 22 \end{cases} \quad (10.2)$$

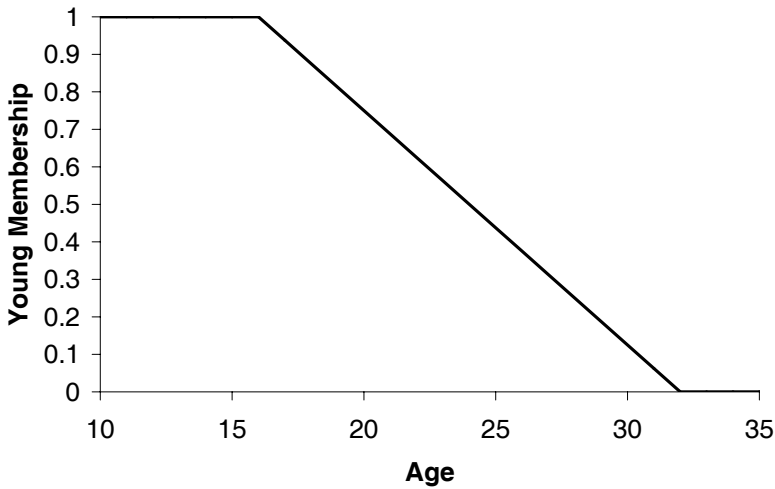


Fig. 10.1 Membership function for the young set.

10.3 Fuzzy Classification Problems

All classification problems we have discussed so far in this chapter assume that each instance takes one value for each attribute and that each instance is classified into only one of the mutually exclusive classes [Yuan and Shaw (1995)].

To illustrate the idea, we introduce the problem of modeling the preferences of TV viewers. In this problem there are three input attributes:

$$A = \{\text{Time of Day, Age Group, Mood}\}$$

and each attribute has the following values:

- $dom(\text{Time of Day}) = \{\text{Morning, Noon, Evening, Night}\}$
- $dom(\text{Age Group}) = \{\text{Young, Adult}\}$
- $dom(\text{Mood}) = \{\text{Happy, Indifferent, Sad, Sour, Grumpy}\}$

The classification can be the movie genre that the viewer would like to watch, such as $C = \{\text{Action, Comedy, Drama}\}$.

All the attributes are vague by definition. For example, peoples feelings of happiness, indifference, sadness, sourness and grumpiness are vague

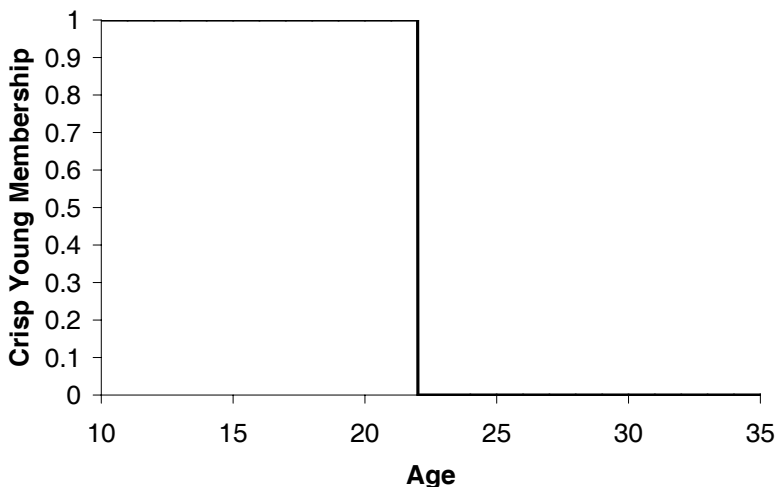


Fig. 10.2 Membership function for the crisp young set.

without any crisp boundaries between them. Although the vagueness of “Age Group” or “Time of Day” can be avoided by indicating the exact age or exact time, a rule induced with a crisp decision tree may then have an artificial crisp boundary, such as “IF Age < 16 THEN action movie”. But how about someone who is 17 years of age? Should this viewer definitely not watch an action movie? The viewer preferred genre may still be vague. For example, the viewer may be in a mood for both comedy and drama movies. Moreover, the association of movies into genres may also be vague. For instance the movie “Lethal Weapon” (starring Mel Gibson and Danny Glover) is considered to be both comedy and action movie.

Fuzzy concept can be introduced into a classical problem if at least one of the input attributes is fuzzy or if the target attribute is fuzzy. In the example described above, both input and target attributes are fuzzy. Formally the problem is defined as following[Yuan and Shaw (1995)]:

Each class c_j is defined as a fuzzy set on the universe of objects U . The membership function $\mu_{c_j}(u)$ indicates the degree to which object u belongs to class c_j . Each attribute a_i is defined as a linguistic attribute which takes linguistic values from $dom(a_i) = \{v_{i,1}, v_{i,2}, \dots, v_{i,|dom(a_i)|}\}$. Each linguistic value $v_{i,k}$ is also a fuzzy set defined on U . The membership $\mu_{v_{i,k}}(u)$ specifies the degree to which object u 's attribute a_i is $v_{i,k}$. Recall that the membership of a linguistic value can be subjectively assigned or

transferred from numerical values by a membership function defined on the range of the numerical value.

10.4 Fuzzy Set Operations

Like classical set theory, fuzzy set theory includes such operations as union, intersection, complement, and inclusion, but also includes operations that have no classical counterpart, such as the modifiers concentration and dilation, and the connective fuzzy aggregation. Definitions of fuzzy set operations are provided in this section.

Definition 10.2 The membership function of the union of two fuzzy sets A and B with membership functions μ_A and μ_B respectively is defined as the maximum of the two individual membership functions $\mu_{A \cup B}(u) = \max\{\mu_A(u), \mu_B(u)\}$.

Definition 10.3 The membership function of the intersection of two fuzzy sets A and B with membership functions μ_A and μ_B respectively is defined as the minimum of the two individual membership functions $\mu_{A \cap B}(u) = \min\{\mu_A(u), \mu_B(u)\}$.

Definition 10.4 The membership function of the complement of a fuzzy set A with membership function μ_A is defined as the negation of the specified membership function $\mu_{\bar{A}}(u) = 1 - \mu_A(u)$.

To illustrate these fuzzy operations, we elaborate on the previous example. Recall that John has a degree of youth of 0.875. Additionally John's happiness degree is 0.254. Thus, the membership of John in the set $\text{Young} \cup \text{Happy}$ would be $\max(0.875, 0.254) = 0.875$, and its membership in $\text{Young} \cap \text{Happy}$ would be $\min(0.875, 0.254) = 0.254$.

It is possible to chain operators together, thereby constructing quite complicated sets. It is also possible to derive many interesting sets from chains of rules built up from simple operators. For example John's membership in the set $\overline{\text{Young}} \cup \text{Happy}$ would be $\max(1 - 0.875, 0.254) = 0.254$

The usage of the max and min operators for defining fuzzy union and fuzzy intersection, respectively is very common. However, it is important to note that these are not the only definitions of union and intersection suited to fuzzy set theory.

10.5 Fuzzy Classification Rules

Definition 10.5 The fuzzy subsethood $S(A, B)$ measures the degree to which A is a subset of B .

$$S(A, B) = \frac{M(A \cap B)}{M(A)} \quad (10.3)$$

where $M(A)$ is the *cardinality* measure of a fuzzy set A and is defined as

$$M(A) = \sum_{u \in U} \mu_A(u) \quad (10.4)$$

The subsethood can be used to measure the truth level of the rule of classification rules. For example given a classification rule such as “IF Age is Young AND Mood is Happy THEN Comedy” we have to calculate $S(Hot \cap Sunny, Swimming)$ in order to measure the truth level of the classification rule.

10.6 Creating Fuzzy Decision Tree

There are several algorithms for induction of decision trees. In this section we will focus on the algorithm proposed by [Yuan and Shaw (1995)]. This algorithm can handle the classification problems with both fuzzy attributes and fuzzy classes represented in linguistic fuzzy terms. It can also handle other situations in a uniform way where numerical values can be fuzzified to fuzzy terms and crisp categories can be treated as a special case of fuzzy terms with zero fuzziness. The algorithm uses classification ambiguity as fuzzy entropy. The classification ambiguity, which directly measures the quality of classification rules at the decision node, can be calculated under fuzzy partitioning and multiple fuzzy classes.

The fuzzy decision tree induction consists of the following steps:

- Fuzzifying numeric attributes in the training set.
- Inducing a fuzzy decision tree.
- Simplifying the decision tree.
- Applying fuzzy rules for classification.

10.6.1 Fuzzifying Numeric Attributes

When a certain attribute is numerical, it needs to be fuzzified into linguistic terms before it can be used in the algorithm. The fuzzification process can be performed manually by experts or can be derived automatically using some sort of clustering algorithm. Clustering groups the data instances into subsets in such a manner that similar instances are grouped together; different instances belong to different groups. The instances are thereby organized into an efficient representation that characterizes the population being sampled.

Yuan and Shaw (1995) suggest a simple algorithm to generate a set of membership functions on numerical data. Assume attribute a_i has numerical value x from the domain X . We can cluster X to k linguistic terms $v_{i,j}, j = 1, \dots, k$. The size of k is manually predefined. For the first linguistic term $v_{i,1}$, the following membership function is used:

$$\mu_{v_{i,1}}(x) = \begin{cases} 1 & x \leq m_1 \\ \frac{m_2-x}{m_2-m_1} & m_1 < x < m_2 \\ 0 & x \geq m_2 \end{cases} \tag{10.5}$$

For each $v_{i,j}$ when $j = 2, \dots, k-1$ has a triangular membership function as follows:

$$\mu_{v_{i,j}}(x) = \begin{cases} 0 & x \leq m_{j-1} \\ \frac{x-m_{j-1}}{m_j-m_{j-1}} & m_{j-1} < x \leq m_j \\ \frac{m_{j+1}-x}{m_{j+1}-m_j} & m_j < x < m_{j+1} \\ 0 & x \geq m_{j+1} \end{cases} \tag{10.6}$$

Finally the membership function of the last linguistic term $v_{i,k}$ is:

$$\mu_{v_{i,k}}(x) = \begin{cases} 0 & x \leq m_{k-1} \\ \frac{x-m_{k-1}}{m_k-m_{k-1}} & m_{k-1} < x \leq m_k \\ 1 & x \geq m_k \end{cases} \tag{10.7}$$

Figure 10.3 illustrates the creation of four groups defined on the age attribute: “young”, “early adulthood”, “middle-aged” and “old age”. Note that the first set (“young”) and the last set (“old age”) have a trapezoidal form which can be uniquely described by the four corners. For example, the “young” set could be represented as $(0, 0, 16, 32)$. In between, all other

sets (“early adulthood” and “middle-aged”) have a triangular form which can be uniquely described by the three corners. For example, the set “early adulthood” is represented as (16, 32, 48).

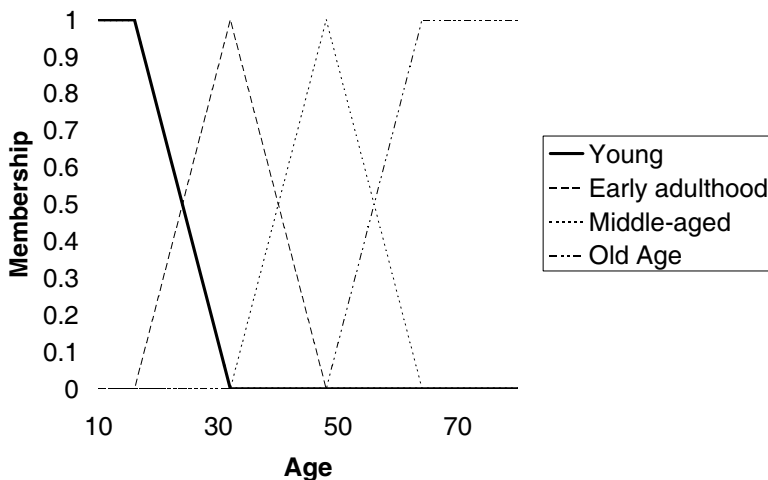


Fig. 10.3 Membership function for various groups in the age attribute.

The only parameters that need to be determined are the set of k centers $M = \{m_1, \dots, m_k\}$. The centers can be found using the algorithm presented in Figure 10.4. Note that in order to use the algorithm, a monotonic decreasing learning rate function should be provided.

10.6.2 Inducing of Fuzzy Decision Tree

The induction algorithm of fuzzy decision tree is presented in Figure 10.5. The algorithm measures the classification ambiguity associated with each attribute and splits the data using the attribute with the smallest classification ambiguity. The classification ambiguity of attribute a_i with linguistic terms $v_{i,j}, j = 1, \dots, k$ on fuzzy evidence S , denoted as $G(a_i|S)$, is the weighted average of classification ambiguity calculated as:

$$G(a_i | S) = \sum_{j=1}^k w(v_{i,j} | S) \cdot G(v_{i,j} | S) \tag{10.8}$$

Require: X - a set of values, $\eta(t)$ - some monotonic decreasing scalar function representing the learning rate.

Ensure: $M = \{m_1, \dots, m_k\}$

- 1: Initially set m_i to be evenly distributed on the range of X .
- 2: $t \leftarrow 1$
- 3: **repeat**
- 4: Randomly draw one sample x from X
- 5: Find the closest center m_c to x .
- 6: $m_c \leftarrow m_c + \eta(t) \cdot (x - m_c)$
- 7: $t \leftarrow t + 1$
- 8: $D(X, M) \leftarrow \sum_{x \in X} \min_i \|x - m_i\|$
- 9: **until** $D(X, M)$ converges

Fig. 10.4 Algorithm for fuzzifying numeric attributes

where $w(v_{i,j} | S)$ is the weight which represents the relative size of $v_{i,j}$ and is defined as:

$$w(v_{i,j} | S) = \frac{M(v_{i,j} | S)}{\sum_k M(v_{i,k} | S)} \tag{10.9}$$

The classification ambiguity of $v_{i,j}$ is defined as $G(v_{i,j} | S) = g(\vec{p}(C | v_{i,j}))$, which is measured based on the possibility distribution vector $\vec{p}(C | v_{i,j}) = (p(c_1 | v_{i,j}), \dots, p(c_{|K|} | v_{i,j}))$.

Given $v_{i,j}$, the possibility of classifying an object to class c_l can be defined as:

$$p(c_l | v_{i,j}) = \frac{S(v_{i,j}, c_l)}{\max_k S(v_{i,j}, c_k)} \tag{10.10}$$

where $S(A, B)$ is the fuzzy subsethood that was defined in Definition 10.5. The function $g(\vec{p})$ is the possibilistic measure of ambiguity or nonspecificity and is defined as:

$$g(\vec{p}) = \sum_{i=1}^{|\vec{p}|} (p_i^* - p_{i+1}^*) \cdot \ln(i) \tag{10.11}$$

where $\vec{p}^* = (p_1^*, \dots, p_{|\vec{p}|}^*)$ is the permutation of the possibility distribution \vec{p} sorted such that $p_i^* \geq p_{i+1}^*$.

All the above calculations are carried out at a predefined significant level α . An instance will take into consideration of a certain branch $v_{i,j}$ only if its corresponding membership is greater than α . This parameter is used to filter out insignificant branches.

After partitioning the data using the attribute with the smallest classification ambiguity, the algorithm looks for nonempty branches. For each nonempty branch, the algorithm calculates the truth level of classifying all instances within the branch into each class. The truth level is calculated using the fuzzy subsethood measure $S(A, B)$.

If the truth level of one of the classes is above a predefined threshold β then no additional partitioning is needed and the node become a leaf in which all instance will be labeled to the class with the highest truth level. Otherwise the procedure continues in a recursive manner. Note that small values of β will lead to smaller trees with the risk of underfitting. A higher β may lead to a larger tree with higher classification accuracy. However, at a certain point, higher values β may lead to overfitting.

Require: S - Training Set A - Input Feature Set y - Target Feature

Ensure: Fuzzy Decision Tree

- 1: Create a new fuzzy tree FT with a single root node.
- 2: **if** S is empty **OR** Truth level of one of the classes $\geq \beta$ **then**
- 3: Mark FT as a leaf with the most common value of y in S as a label.
- 4: Return FT .
- 5: **end if**
- 6: $\forall a_i \in A$ find a with the smallest classification ambiguity.
- 7: **for** each outcome v_i of a **do**
- 8: Recursively call procedure with corresponding partition v_i .
- 9: Connect the root node to the returned subtree with an edge that is labeled as v_i .
- 10: **end for**
- 11: Return FT

Fig. 10.5 Fuzzy decision tree induction

10.7 Simplifying the Decision Tree

Each path of branches from root to leaf can be converted into a rule with the condition part representing the attributes on the passing branches from the root to the leaf and the conclusion part representing the class at the leaf with the highest truth level classification. The corresponding classification rules can be further simplified by removing one input attribute term at a time for each rule we try to simplify. Select the term to remove with the highest truth level of the simplified rule. If the truth level of this new rule is not lower than the threshold β or the truth level of the original rule, the simplification is successful. The process will continue until no further simplification is possible for all the rules.

10.8 Classification of New Instances

In a regular decision tree, only one path (rule) can be applied for every instance. In a fuzzy decision tree, several paths (rules) can be applied for one instance. In order to classify an unlabeled instance, the following steps should be performed [Yuan and Shaw (1995)]:

- Step 1: Calculate the membership of the instance for the condition part of each path (rule). This membership will be associated with the label (class) of the path.
- Step 2: For each class calculate the maximum membership obtained from all applied rules.
- Step 3: An instance may be classified into several classes with different degrees based on the membership calculated in Step 2.

10.9 Other Fuzzy Decision Tree Inducers

There have been several fuzzy extensions to the ID3 algorithm. The UR-ID3 algorithm [Maher and Clair (1993)] starts by building a strict decision tree, and subsequently fuzzifies the conditions of the tree. Tani and Sakoda (1992) use the ID3 algorithm to select effective numerical attributes. The obtained splitting intervals are used as fuzzy boundaries. Regression is then used in each subspace to form fuzzy rules. Cios and Sztandera (1992) use the ID3 algorithm to convert a decision tree into a layer of a feedforward neural network. Each neuron is represented as a hyperplane with a fuzzy

boundary. The nodes within the hidden layer are generated until some fuzzy entropy is reduced to zero. New hidden layers are generated until there is only one node at the output layer.

Fuzzy-CART [Jang (1994)] is a method which uses the CART algorithm to build a tree. However, the tree, which is the first step, is only used to propose fuzzy sets of the continuous domains (using the generated thresholds). Then, a layered network algorithm is employed to learn fuzzy rules. This produces more comprehensible fuzzy rules and improves the CART's initial results.

Another complete framework for building a fuzzy tree including several inference procedures based on conflict resolution in rule-based systems and efficient approximate reasoning methods was presented in [Janikow, 1998].

Olaru and Wehenkel (2003) presented a new type of fuzzy decision trees called soft decision trees (SDT). This approach combines tree-growing and pruning, to determine the structure of the soft decision tree. Refitting and backfitting are used to improve its generalization capabilities. The researchers empirically showed that soft decision trees are significantly more accurate than standard decision trees. Moreover, a global model variance study shows a much lower variance for soft decision trees than for standard trees as a direct cause of the improved accuracy.

Peng (2004) has used FDT to improve the performance of the classical inductive learning approach in manufacturing processes. Peng proposed using soft discretization of continuous-valued attributes. It has been shown that FDT can deal with the noise or uncertainties existing in the data collected in industrial systems.