Belief Decision Trees: Theoretical foundations

Zied Elouedi[1], Khaled Mellouli[1], Philippe Smets[2]
[1] Institut Supérieur de Gestion de Tunis,
41 Avenue de la liberté, cité Bouchoucha,
2000 Le Bardo, Tunis, Tunisia
[2] IRIDIA, Université Libre de Bruxelles,
50 av., F. Roosvelt, CP194/6, 1050 Brussels, Belgium
zied.elouedi@isg.rnu.tn, khaled.mellouli@ihec.rnu.tn,
psmets@ulb.ac.be

September 6, 2000

Abstract

This paper extends the decision tree technique to an uncertain environment where the uncertainty is represented by belief functions as interpreted in the Transferable Belief Model (TBM). This so-called belief decision tree is a new classification method adapted to uncertain data. We will be concerned with the construction of the belief decision tree from a training set where the knowledge about the instances' classes is represented by belief functions, and its use for the classification of new instances where the knowledge about the attributes' values is represented by belief functions.

Keywords: Belief Functions, Decision Tree, Belief Decision Tree, Classification, Transferable Belief Model.

1 Introduction

Several learning methods have been developed to ensure classification. Among these, the decision tree method may be one of the most commonly used in supervised learning approaches. Indeed decision trees are characterized by their capability to break down a complex decision problem into several simpler ones. They represent a sequential procedure for deciding the class membership of a given instance. Their major advantage resides in providing a powerful formalism for representing comprehensible classifiers often easy to interpret by experts and even by ordinary users.

Though their popularity and efficiency, the standard decision trees are unfit to cope with data pervaded with uncertainty both at the construction and at the classification phase. Ignoring the uncertainty may affect classification results and even produce erroneous decisions.

In order to overcome these limitations encountered in standard decision trees, two kinds of technique have been proposed: **the probabilistic decision trees** [23] and **the fuzzy decision trees** [41] [44] [18].

The major objective of the probabilistic decision trees is to classify instances with missing or uncertain attribute values where the uncertainty is represented by a probability.

The fuzzy decision trees have been developed to cope with data where both the class and the attributes are represented by fuzzy values. Fuzziness that occurs in classes and attributes are not considered in this paper, but the technique developed here could be extended to such types of data.

The belief function theory, as understood in the Transferable Belief Model [35], [37], provides a very powerful tool to deal with epistemic uncertainty. It provides a mathematical tool to treat subjective, personal judgments on the different parameters of any classification problem and can be easily extended to deal with objective probabilities. It allows experts to express partial beliefs in a much more flexible way than probability functions do. It permits to handle nicely partial or even total ignorance concerning classification parameters.

Besides, it offers interesting tools to combine several pieces of evidence, like the conjunctive and the disjunctive rules of combination. In addition, decision making is solved through the pignistic transformation.

Hence, the belief function theory seems to provide a convenient framework to handle uncertainty in the decision tree techniques. For this reason, we develop what we call **a belief decision tree approach** which will integrate the advantages of both the decision tree technique and the belief function theory in order to deal with the uncertainty, especially the cognitive one, that may affect the classification problem parameters.

This paper presents basically the theoretical basis of the belief decision trees. It is organized as follows: we start by giving the necessary background concerning the decision trees and the belief function theory. The characteristics of the belief decision trees are then defined in section 4, whereas in section 5 the parameters of this new classification method are developed. Finally in section 6 and section 7, we detail the construction of a belief decision tree and the classification procedures.

This paper focuses on theoretical foundations of the belief decision trees. Its application and its comparison with other decision trees will be presented in a forthcoming paper. Preliminary results have been published in [8, 9]. Another approach to build belief decision trees is presented in [2, 7] where they consider the data as a "random sample" but are limited to binary classes, whereas we use an other interpretation of the data as explained in section 5.

2 Decision trees

A decision tree is a representation of a decision procedure allowing to determine the class of an object. It is composed of three basic elements:

1. **A decision node** specifying the test attribute.
2. **An edge** corresponding to one of the possible values of the test attribute outcomes. It leads generally to a sub decision tree.
3. **A leaf**, which is also named **an answer node**, including classes that, typically, belongs to the same class, or at least are very similar.

For what concerns a decision tree, the developer must explain how the tree is constructed and how it is used.

1. **Building the tree.** Based on a given training set, a decision tree is built. It consists in selecting for each decision node the "appropriate" test attribute and also to define the class labeling each leaf.
2. **Classification.** Once the tree is constructed, it is used in order to classify a new instance. We start at the root of the decision tree, we test the attribute specified by this node. The result of this test allows us to move down the tree branch according to the attribute value of the given instance. This process is repeated until a leaf is encountered, the instance is being then classified in the same class as the one characterizing the reached leaf.

Several algorithms have been developed in order to ensure the construction of decision trees and its use for the classification task. The **ID3** and **C4.5** algorithms developed by Quinlan [20], [24] are probably the most popular ones. We can also mention the **CART** algorithm of Breiman and al. [3].

The majority of the algorithms for building decision trees use a descendent strategy (from the root to the leaves). To ensure this approach, many parameters are required and they can be considered as generic parameters of the algorithm.

The formalism for building decision trees is also referred to as Top Down Induction of Decision Tree (TDIDT) since it proceeds by successive divisions of the training set where each division represents a question about an attribute value.

A generic decision tree algorithm is characterized by the next properties.

1. **The attribute selection measure.** An attribute is chosen in order to partition the training set in an *"optimized"* manner. A decision node relative to this attribute is created. It becomes the root of the corresponding decision tree.
2. **A partitioning strategy.** The current training set is divided by taking into account the selected test attributes.

3. **The stopping criteria.** A training subset is declared as a leaf if it satisfies one of the stopping criteria.

The different steps are applied recursively on the training subsets that do not verify the stopping criteria.

Attribute selection measure: This measure allows to select the attribute that characterizes the root of the decision tree and those of the different sub decision trees. Quinlan has defined a measure called information gain [20]. It is also referred to as the 'gain criterion' based on the information theory of Shannon [29]. The idea is to compute the information gain of each attribute in order to find how well each attribute alone classifies the training examples, then the one presenting the highest value will be chosen. In fact, this attribute generates a partition where the instances classes are as homogeneous as possible within each subset created by the attribute.

Let T denote a training set. Let $\Theta = \{C_1, C_2 \ldots, C_n\}$ be the set of n mutually exclusive and exhaustive classes so that each instance in T belongs to one and only one class. Let A be one attribute which domain is finite and denoted by $D(A)$. The *information gain criterion* of Quinlan is defined as following [24]:

$$Gain(T, A) = Info(T) - Info_A(T) \tag{1}$$

$$\text{where } Info(T) = -\sum_{i=1}^{n} \frac{freq(C_i, T)}{|T|}.log_2 \frac{freq(C_i, T)}{|T|} \tag{2}$$

$$\text{and } Info_A(T) = \sum_{v \in D(A)} \frac{|T_v|}{|T|}.Info(T_v) \tag{3}$$

where $freq(C_i, T)$ denotes the number of objects in the set T that belong to the class C_i and T_v is the subset of objects for which the attribute A has the value v.

The best attribute is the one that maximizes $Gain(T, A)$. Once the best attribute is allocated to a node, the training set T is split into several subsets, one for each value of the selected attribute. The procedure is then iterated for each subset using only the data that belong to them.

Although it has shown good results, the information gain criterion presents a serious limitation. It favors attributes with a large number of values over those with a small number of values [24]. To overcome this drawback, Quinlan has proposed a kind of normalization known as *the gain ratio criterion.* In this manner, the attributes with many values will be adjusted.

$$Gain\ ratio(T, A) = \frac{Gain(T, A)}{Split\ Info(T, A)} \tag{4}$$

$$\text{where } Split\ Info(T, A) = -\sum_{v \in D(A)} \frac{|T_v|}{|T|} log_2 \frac{|T_v|}{|T|} \tag{5}$$

$Split\ Info(T, A)$ measures the information in the attribute due to the partition of the training set T into $|D(A)|$ training subsets. This quantity describes the information content of the attribute itself.

The idea is to compute the gain ratio of each test attribute, the one presenting the highest value will be selected as the attribute test.

Partitioning strategy: It consists in decomposing the training set into many subsets. In the case of symbolic attributes (with a finite number of values), this strategy resides on testing all the possible attribute values, whereas for the case of numeric attributes, a discretization step is generally needed.

Stopping criteria: It deals with the condition of stopping growth of a part of the decision tree (or even all the decision tree). In other words, it determines whether or not a training subset will be further divided. It is generally fulfilled when all the objects belong to only one class. The part of the decision tree verifying this criterion will be declared as a leaf.

The problems of overfitting and tree pruning are not considered in this paper.

3 Belief function theory

3.1 Basic concepts

In the following, we shall briefly recall some of the basics of the belief function theory. Details can be found in [26, 33, 38].

Let Θ be a finite non empty set including all the elementary events related to a given problem. In the present context, Θ is a set of classes, and the elementary events are the possible classes. These events are assumed to be exhaustive and mutually exclusive. Such set Θ is classically called the frame of discernment.

The power set of Θ, denoted by 2^Θ, is defined as:

$$2^\Theta = \{A : A \subseteq \Theta\} \tag{6}$$

Hence, the elements of 2^Θ are sets of classes.

The impact of a piece of evidence on the different subsets of the frame of discernment Θ is represented by the so-called basic belief assignment (bba), (initially called basic probability assignment [26], an expression that has created confusion in the past). The bba m is a function $m : 2^\Theta \rightarrow [0, 1]$ that satisfies:

$$\sum_{A \subseteq \Theta} m(A) = 1 \tag{7}$$

The value $m(A)$, called a basic belief mass (bbm), represents the part of belief exactly committed to the subset A of Θ given a piece of evidence, or equivalently to the fact that all we know is that A holds. Due to the lack of information, this quantity cannot be apportioned to any strict subset of A. So, it represents the specific support given to A.

Shafer [26] has initially proposed a normality condition expressed by:

$$m(\emptyset) = 0 \tag{8}$$

A bba that satisfies (8) is called a normalized basic belief assignment function.

Smets [32, 37] relaxes this condition and interprets $m(\emptyset)$ as the part of belief given to the fact that none of the hypotheses in Θ is true or as the amount of conflict between the pieces of evidence. The subsets A of the frame of discernment Θ such that $m(A)$ is strictly positive, are called the **focal elements** of the bba m.

A belief function, denoted bel, corresponding to a specific bba m, assigns to every subset A of Θ the sum of the basic belief masses committed to the subsets of A by m [26]. Contrary to the bba which expresses only the part of beliefs committed exactly to A, the belief function bel represents the total belief that one commits to A without being also committed to $\overline{A}$. The belief function $bel : 2^\Theta \to [0, 1]$ is defined so that:

$$bel(A) = \sum_{\emptyset \neq B \subseteq A} m(B), \ \forall A \subseteq \Theta. \tag{9}$$

The plausibility function $pl : 2^\Theta \to [0, 1]$ quantifies the maximum amount of belief that could be given to a subset A of Θ. It is equal to the sum of the bbm given to subsets B compatible with A. In other words, it contains those parts of beliefs that do not contradict A (i.e., those B such that $B \cap A \neq \emptyset$).

$$pl(A) = \sum_{A \cap B \neq \emptyset} m(B), \ \forall A \subseteq \Theta. \tag{10}$$

We also deal with the so-called commonality function $q : 2^\Theta \to [0, 1]$. It represents the total mass free to move to every element of A [1]. It is defined as:

$$q(A) = \sum_{A \subseteq B} m(B), \ \forall A \subseteq \Theta. \tag{11}$$

Several special belief functions are described.

- **The vacuous belief function** is a belief function that satisfies [26]:

  $$m(\Theta) = 1 \text{ and } m(A) = 0, \ \forall A \subseteq \Theta, A \neq \Theta. \tag{12}$$

  Such bba quantifies the state of **total ignorance** since there is no support given to any strict subset of Θ.

- **A categorical belief function** is a belief function that satisfies [19]:

  $$m(A) = 1 \text{ for some } A \subseteq \Theta, A \neq \Theta \text{ and } m(B) = 0, \ \forall B \subseteq \Theta, B \neq A. \tag{13}$$

  Such function has a unique focal element A different from the frame of discernment Θ.

- **The certain belief function** is a categorical belief function which focal element is a singleton. It represents a state of total certainty as it assigns all the belief to a unique elementary event.

3.2 Combination of belief functions

Let m_1 and m_2 be two bba's defined on the same frame of discernment Θ. These two bba's are induced by two 'distinct' pieces of evidence and collected from two experts (information sources). These bba's can be combined either conjunctively or disjunctively:

1. **The conjunctive rule:** The conjunctive rule of combination handles the case where both sources of information are fully reliable. The result of the combination is a joint bba representing the conjunction of the two pieces of evidence induced from the two sources. Hence, the induced bba quantifies the combined impact of the two pieces of evidence. This rule, denoted by $\bigcirc\!\!\!\!\cap$, is defined by [39]:

$$(m_1 \bigcirc\!\!\!\!\cap m_2)(A) = \sum_{B,C \subseteq \Theta : B \cap C = A} m_1(B).m_2(C) \tag{14}$$

   The conjunctive rule can be seen as an unnormalized Dempster's rule of combination. This latter, denoted by $\oplus$, deals with the closed world assumptions [33], and is defined as [26]:

$$(m_1 \oplus m_2)(A) = K. \sum_{B,C \subseteq \Theta : B \cap C = A} m_1(B).m_2(C) \tag{15}$$

   where

$$K^{-1} = 1 - \sum_{B,C \subseteq \Theta : B \cap C = \emptyset} m_1(B).m_2(C) \tag{16}$$

$$\text{and } (m_1 \oplus m_2)(\emptyset) = 0 \tag{17}$$

   K is called **the normalization factor**.

2. **The disjunctive rule:** The dual of the conjunctive rule is the disjunctive rule of combination that builds the bba representing the impact of two pieces of evidence when we only know that at least one is to be accepted, but we do not know which one. This rule, denoted by $\bigcirc\!\!\!\!\cup$, is defined by [39]:

$$(m_1 \bigcirc\!\!\!\!\cup m_2)(A) = \sum_{B,C \subseteq \Theta : B \cup C = A} m_1(B).m_2(C). \tag{18}$$

Note that since the conjunctive and the disjunctive rules are both commutative and associative, combining several pieces of evidence induced from distinct information sources (either conjunctively or disjunctively) may be easily ensured by applying repeatedly the chosen rule. The conjunctive and disjunctive rules are not distributive.

3.3 Vacuous extension of belief functions

Let X and Y be two sets of variables such that Y $\subseteq$ X. Let m^Y be a bba defined on the domain Θ_Y of Y. The vacuous extension of m^Y to Θ_X, denoted $\mathrm{m}^{Y\uparrow X}$, is the bba obtained by extending the information in m^Y to a larger frame X [4]:

$$m^{Y\uparrow X}(A \times \Theta_{X-Y}) = m^Y(A) \text{ for } A \subseteq \Theta_Y \tag{19}$$

$$m^{Y\uparrow X}(B) = 0 \text{ if } B \text{ is not in the form } A \times \Theta_{X-Y} \tag{20}$$

3.4 Pignistic transformation

The problem of decision making in the context of the Transferable Belief Model is handled by the pignistic transformation. The TBM is based on a two level mental models:

- *The credal level* where beliefs are entertained and represented by belief functions.
- *The pignistic level* where beliefs are used to make decisions and represented by probability functions called the pignistic probabilities.

When a decision must be made, beliefs held at the credal level induce a probability measure at the pignistic level, measure denoted $BetP$ [38]. The link between these two functions is achieved by the pignistic transformation.

$$BetP(A) = \sum_{B \subseteq \Theta} \frac{|A \cap B|}{|B|} \frac{m(B)}{1 - m(\emptyset)}, \; \forall A \subseteq \Theta \tag{21}$$

4 Characteristics of a belief decision tree

In this section, we introduce the concept of the belief decision tree by presenting its definition and its objectives. Then, we define the structure of the training set which will be illustrated by an example. Finally, the belief decision tree representation will be described.

4.1 Definition

A belief decision tree is a decision tree in an uncertain environment where the uncertainty is represented by belief functions as interpreted in the Transferable Belief Model (TBM).

Contrary to a classical decision tree where the objects' classes and the attribute values are known with certainty, in a belief decision tree, these two parameters may be uncertain or even unknown. Such uncertainty can appear either in the construction or in the classification phase.

4.2 Objectives

As for a standard decision tree, a belief decision tree aims at realizing two major objectives:

1. Building a decision tree from a given set of training instances pervaded with uncertainty. In other words, ensuring the induction of the belief decision tree.

2. Ensuring the classification of new instances which attributes' values can be uncertain or even unknown. Such procedure is also called the inference procedure.

4.3 Structure of the training set

4.3.1 Definition

Any decision tree is constructed from a training set of objects and based on successive refinements. The training set is the basis leading to the induction of the tree and consequently to the classification of new instances in the inference phase.

This set is traditionally composed of elements represented as pairs (attributes, class) where for each object, we know exactly its assigned class and the value of each attribute.

However, due to the uncertainty introduced here, the structure of the training set we will use may be different from the traditional one. We accept that it may contain data where the class may be uncertain or even unknown. We assume that the values of the attributes of each training instance are known with certainty. Uncertainty on the attributes' values of the cases in the training set is not considered in this paper.

The uncertainty on the classes of the training instance is represented by a basic belief assignment defined on the set of possible classes. This bba, generally given by an expert (or several experts), represents the opinions-beliefs of this expert about the actual value of the class for each object in the training set.

4.3.2 Notations

In this paper, we use the following notations:
- T: a given training set,
- I_j: an instance also named object or case or example,
- $\mathbf{A}$: the set of attributes,
- $\Theta = \{C_1, C_2, ..., C_n\}$: the frame of discernment involving all the possible classes

related to the classification problem. The C_i's are assumed to be mutually exclusive and exhaustive,
- C(I_j): the actual class of the object I_j,
- $m_g^{\Theta}\{I_j\}[a](C)$: the conditional bbm given to the hypothesis that the actual class $C(I_j)$ of instance I_j belongs to $C \subseteq \Theta$ by an agent g that accepts that the information a is true. Useless indices are omitted when they are clearly defined from the context.

4.3.3 Special cases

Among the advantages of working under the belief function framework, we notice that the two extreme cases, total knowledge and total ignorance concerning training instance classes, are easily expressed:

- When the class $C(I_j)$ of the object I_j is perfectly known to be $C_\nu \in \Theta$, it will be represented by a certain belief function having this class as the only focal element:

$$m^{\Theta}(C_\nu) = 1 \text{ and } m^{\Theta}(C) = 0 \text{ for all } C \neq C_\nu, \quad C \subseteq \Theta \tag{22}$$

  where $C_\nu = C(I_j)$ is a singular class.

  Such case is referred to as **total knowledge** and corresponds to the classical *"certain"* context. Hence, our representation is also appropriate (but not efficient) to describe the standard case where all the classes of the training instances are known with certainty.

- When we have no information about the class of an object which means that the expert is not able to give any judgment concerning the instance's classes, then the bba will be a vacuous belief function defined by:

$$m^{\Theta}(\Theta) = 1 \text{ and } m^{\Theta}(C) = 0 \text{ for } C \subset \Theta \tag{23}$$

  Such a case is referred to as **total ignorance**. These cases are generally not included in a training set since they do not provide any information regarding the instances' classes.

- In addition to these special cases, the case of disjunctive classes may be easily described by the so-called categorical belief function. In practice, this latter case may often happen and corresponds to a disjunction of classes that will be assigned to a training instance by an expert. It represents probably the most typical type of information where belief functions will be applied.

  Such opinion will be represented by a bba characterized by only one focal element representing the union of the classes assigned to this training instance.

4.3.4 Example

This is a simple example to illustrate our structure of the training set T within the belief function framework. Let T be a small training set (see Table 1) composed of eight instances. Each training instance is described by three symbolic attributes.

- Hair with possible values {Blond, Red, Dark}.
- Eyes with possible values {Blue, Brown}.
- Height with possible values {Tall, Short}.

The set of the possible classes are: $\Theta = \{C_1, C_2, C_3\}$.

Hair	**Eyes**	**Height**	**bba on classes**	C_1	C_2	C_3
Blond	Blue	Tall	$C_1, .8; \Theta, .2$	.86	.07	.07
Blond	Brown	Tall	$C_2, .4; C_1 \cup C_2, .4; \Theta, .2$	.27	.66	.07
Blond	Blue	Tall	$C_1, .9; \Theta, .1$	.94	.03	.03
Blond	Brown	Short	$C_2, .6; C_3, .2; \Theta, .2$	.07	.66	.27
Red	Blue	Tall	$C_2, .8; \Theta, .2$	.07	.86	.07
Dark	Brown	Short	$C_3, .6; \Theta, .4$	.13	.13	.74
Dark	Brown	Tall	$C_3, .9; \Theta, .1$	.03	.03	.94
Dark	Brown	Tall	$C_3, .5; C_1 \cup C_3, .2; \Theta, .3$	.20	.10	.70

Table 1: Training set T. Values of the attributes for the eight cases, and the bba's on the classes (represented by pairs which components are the focal element and its bbm). The last three columns give the pignistic probabilities on the three classes.

So for case 1, the expert strongly supports (.8) that it is a C_1 case. For case 2, C_2 seems supported but C_1 is not excluded, etc... Methods to assess these bba's are explained in [39].

4.4 Belief decision tree representation

Once the structure of the training set is defined, the representation of our belief decision tree is composed of the same elements as in the traditional decision tree:

- **Decision nodes** for testing attributes.
- **Branches** for specifying attribute values.
- **Leaves** dealing with classes of the training instances.

Due to the uncertainty related to training instances' classes, the structure of leaves will change. Instead of assigning a unique class to each leaf, it will be labeled by a bba expressing a belief on the actual class of the objects belonging to the leaf. The computation of each leaf's bba will be shown in the next section.

5 The averaging and the conjunctive approaches for building belief decision trees

The training set in a belief decision tree is characterized by the fact that our knowledge about the value of the actual class of its instances is represented by a bba on Θ, the set of possible classes. The classical algorithms must be adapted to cope with such a context poisned with uncertainty. Before describing our algorithms in details, we explain how we derive two of them. Of course, these are many possible algorithms, but the interest of the chosen ones comes from their close link to:

- the classical approach developed by Quinlan for our so-called averaging approach
- the ideas behind the TBM itself for our so-called conjunctive approach.

5.1 The averaging approach

Suppose the decision tree is built, and consider one leaf, denoted S, and suppose that 80% of the cases in S are C_1 cases. Why does this influence our knowledge about the class to which belongs a new case that falls in the same leaf S ? There are (at least) two possible answers.

5.1.1 The sampling answer.

We implicitly consider that the new case is a case selected at random from the same population as the one from which the cases in S were selected. We assume that the observed frequencies in S are 'good estimators' of the distribution of the classes in the population. Then the proportion 80% is equated (maybe in a not very rigorous way) to the probability that a case randomly selected in the population represented by S is C_1, i.e., belongs to class C_1. We observe a new case, and we can say that the probability that the new case is a C_1 is .80. Therefore the 80% proportion of C_1 in S becomes finally the probability that the new case that falls in S is a C_1.

The largest the dominant proportion in S, the most confident we will feel in our claim about the new case's class. So we would like that most cases in a leaf belong to the same class. If this ideal is not achievable, we would like to be able to assert with confidence that the new case's class is one among a few of the possible classes, and that many of the possible classes can be excluded. The worst case for a leaf is encountered when every class is equally represented, as assigning then a class to a new case falling in that leaf would be unjustified and unsupported.

The heterogeneity of the probabilities is what entropy of equation (2) is supposed to quantify. The entropy is maximal when the classes are equiprobable, and becomes smaller when the distribution of the classes becomes further and further away from the equi-probability. The smallest entropy (of value 0) is reached when the probability is one for one class, and 0 for all the other classes.

Quinlan's algorithm is based on that idea, and thus tries to minimize the entropy at the leaf's level. Unfortunately the justification for using the proportions instead of the probabilities can seriously be criticized. It would be acceptable if the number of cases in S was really large, but in practice this is not the case. Leaves with one element are even considered. So the suggested justification is hard to defend, and we feel the second one is more appropriate.

5.1.2 The finite population answer.

Suppose that the data of S correspond to a population, not to a sample selected from a larger population. We assume that one case is selected at random, with equi-probability, i.e., with probability $1/|S|$, from S, and that the new case is a duplicate of this selected case, so it is exactly equal to the selected one. If we knew which case had been selected, the class of the new case would be the class of the selected one, but we do not know which one in S was actually selected. All we can say is that the probability that the new case's class is C_i is equal to the probability that the selected case is a C_i, and this probability is equal, thanks to the equi-probable sampling method, to the proportions of C_i cases in S.

Entropy becomes then perfectly meaningful for the same reasons as given in the previous analysis.

case i	p_i	C_1	C_2	C_3
1	.2	1	0	0
2	.2	1	0	0
3	.2	0	1	0
4	.2	0	1	0
5	.2	0	0	1
mean		2/5	2/5	1/5

Table 2: Training subset S (Standard case)

For instance, let table 2 represents the five cases in the subset S of the training set. The class of each case is defined by the indicator function. Cases 1 and 2 are C_1's, etc ... If each case has a probability $p_i = .2$ of being selected, then the probability of selecting a C_1 case is just the sum of the indicators weighted by the .2 probabilities. In fact the indicator can be understood as the probability that the selected case is a C_j case given the selected case is case i. So the probability that the selected case is C_j becomes:

$$P(C_j) = \sum_{i \in S} P(C_j | \text{ selected case is } i)\ p_i. \tag{24}$$

Entropy could then be computed from these 'expected values'.

This equation (24) allows us to shift directly to the case where the classes are uncertain, and *the uncertainty is represented by a probability measure*. Table 3 presents the kind of data that could be collected from the five cases in S. Here case 1 has a high probability of being a C_1, but might also be a C_2 or a C_3, even though these last two options are individually less probable than the first. The expected values are computed as in the previous cases, and the probability that the randomly selected case is a C_1 is .2 in this case. Entropy could then be computed from these 'expected value'.

What about the entropy in this context? We would like that there would be as few ambiguity as possible when we classify a new case falling in a leaf. So we would like that the probability in a leaf points essentially to one class, and entropy is an excellent measure to quantify this tendency. Hence the use of the entropy computed from the average probability function in a leaf is plainly justified.

case i	p_i	C_1	C_2	C_3
1	.2	.5	.2	.3
2	.2	.4	.6	0
3	.2	0	1	0
4	.2	.1	.7	.2
5	.2	0	.5	.5
mean		1/5	3/5	1/5

Table 3: Training subset S (Probabilistic case)

Suppose now that *the uncertainty is represented by a bba*, with m_i the bba about the actual class to which case i belongs. If one ask what is the bba of the class to which belongs the randomly selected case, the answer happens to be the average of the m_i's. This results from the fact the bba's m_i are just conditional bba's and if a case is selected according to a probability distribution, then the resulting bba, denote $\overline{m}$, is :

$$\overline{m} = \sum_{i \in S} m_i \; p_i$$

This formula results from the conjunctive combination of the probability function p_i and the conditional bba's m_i's, followed by a marginalization of Θ. It uses the relation [34]: $m_{1 \circledcirc 2}(A) = \sum_{B \subseteq \Omega} m_1[B](A) \; m_2(B)$, where m_1 and m_2 are defined on Ω. The p_i's are the m_2, and the m_i's are the $m_1[B]$ where $B = \{(i, C_j) : C_j \in \Theta\}$, $\Omega = S \times \Theta$, and $A \subseteq \Theta$.

The probability used to compute the entropy is replaced now by the pignistic probability computed from $\overline{m}$. It just happens that the pignistic probability computed form $\overline{m}$ is the same as the average of the pignistic probabilities computed for each case:

$$BetP = \Gamma(\sum_{i \in S} m_i \; p_i) = \sum_{i \in S} \Gamma(m_i) \; p_i.$$

where Γ is the operator that transforms a bba into a pignistic probability function. This linearity property is even the major property that justifies the used of the pignistic probabilities [38]. In consequence, when uncertainty is represented by bba's, it is enough to compute the pignistic probability over Θ, the set of possible classes, for each case, and proceeds as done in Table 3. The use of the entropy at the leaf level is justified just as in the probabilistic case.

Using bba's instead of probabilities on the classes is really not a real issue. One may then raise the question : why to use the TBM in such a case ? The answer is to be found in dynamic contexts where the beliefs about the classes for each individual can vary with time. New pieces of information could be collected about the data in the training set, in which case the bba's will be adapted by applying the appropriate Dempster's rules, and the change in the pignistic probabilities will not be those one obtained if the impact of the new information was handled within the probability model. So using the TBM, eventhough not essential when the training set is fixed, becomes interesting when our knowledge about the classes of the data in the training set can vary.

Besides, suppose the value of a needed attribute of a new case is itself uncertain, like the value being v_1 or v_2. As explained in Section 8, we will compute the bba m_1 from the data in the leaf reached if v_1 was the case, and the bba m_2 from the data in the leaf reached if v_2 was the case. The combination of these two bba's is obtained by the disjunctive rule of combination, something that brings us far away from the pignistic probabilities, and requires the whole TBM apparatus.

So even though in simple cases, the need for the TBM was not essential, it becomes really so in more complicated contexts.

5.2 The conjunctive approach

The second method we considered is conceptually much closer to the TBM itself. Let us first reconsider what can be done at the leaf's level. In a given leaf S, we suppose we have several cases and ideally they all belong to the same class. So every case in S belongs to the same class, but we don't know which one. Each case provides a bba m_i, $i = 1 \ldots |S|$, that represents what is known from case i about the class to which it belongs, hence to the class that characterizes S. The belief we can build on the class 'common' to those who belong to S is obtained by combining the m_i, for $i \in S$, by the conjunctive combination rule. This idea is based on similar approaches developed by Denoeux [5], [40].

So if m_i is the bba of case I_i in the considered leaf S, we compute

$$m_S = \textcircled{\cap}_{i \in S} m_i.$$

The bba m_S is what all the cases in leaf S jointly express about the class of those cases that belong to the leaf S.

So the classification of a new case that falls in leaf S is based on this joint bba m_S, and the class is decided using the pignistic probabilities computed from this bba m_S.

Knowing what will be done once the tree is built, let us now shift to its construction. What 'nice' property should be satisfied by the cases in a leaf. Ideally, they should belong to the same class. But their actual classes are unknown. Suppose then two cases with the same bba on Θ, the set of possible classes. It seems reasonable to be satisfied if both cases fall in the same leaf. So what we would like is that all cases in a leaf have bba's that are 'close' to each others. Thus a distance between bba's, and in particular between two bba's, is required.

5.2.1 Distance between bba's

Let m_1 and m_2 be two bba's, both defined on Θ. These two bba's are vectors in a $2^{|\Theta|}$ dimension space. A natural distance is the euclidian distance between the two vectors. But why to use the bba's themselves, and not any vector that is in one to one correspondence with the bba, like the *bel* vector, or the q vector... So let f_i be such a vector where f_i is a function of m_i, which value at $X \subseteq \Theta$ is denoted $f_i(X)$. We are going to show that $f_i(X) = -ln(q_i(X))$ is an appropriate choice. We define the distance between two instances I_i and I_j belonging to S as:

$$d_{ij}^2 = \sum_{X \subseteq \Theta} (f_i(X) - f_j(X))^2.$$

We can then define the distance among the instance within one group S as the average of the distance between pairs of instances in S:

$$D_S^2 = \frac{1}{2s^2} \sum_{i,j \in S} d_{ij}^2.$$

where $s = |S|$.

Ideally this distance should be minimized if the goal is that cases in S are similar to each others. This 'intra-group' distance (because computed within one group), has the advantage that minimizing it is equivalent to maximizing the 'inter-groups' distances (the one computed between groups), another criterion that could have been advocated.

The intra-group distance can be shown to be equal to:

$$
\begin{aligned}
D_S^2 &= \frac{1}{2s^2}\sum_{ij\in S}\sum_X (f_i(X)-f_j(X))^2 \\
&= \frac{1}{2s^2}\sum_X\sum_{ij\in S}(f_i(X)^2-2f_i(X)f_j(X)+f_j(X)^2) \\
&= \frac{1}{2s^2}\sum_X(2s\sum_{i\in S}f_i(X)^2-2(\sum_{i\in S}f_i(X))^2) \\
&= \frac{1}{s}\sum_X(\sum_{i\in S}f_i(X)^2-\frac{1}{s}(\sum_{i\in S}f_i(X))^2\ ) \\
&= \sum_X\frac{1}{s}\sum_{i\in S}(\ f_i(X)-\overline{f(X)}\ )^2 \\
&\propto \sum_X \text{variance}(f_i(X))
\end{aligned}
$$

The distance D_S^2 depends thus of $\overline{f(X)}$. This average $\overline{f}$ can be seen as the function of a bba $\overline{m}$ that has the following property: when 'added' s times, the result has the same 'weight' as the 'addition' of the s bba's m_i. As far as within one group, all bba's will be summarized by the result of their combination by the conjunctive rule of combination, the 'addition' operation can be seen as applying the conjunctive combination. So we want:

$$\bigcirc\!\!\!\!\cap_{i=1\ldots s}\overline{m} = \bigcirc\!\!\!\!\cap_{i=1\ldots s}\ m_i.$$

Thanks to the property of the commonality functions, the conjunctive combination rule can be written as a product, and even better, the logarithm of the commonality function of the combination is the sum of the logarithms of the commonality functions entered into the conjunctive combination.

Let κ be defined as minus the logarithm (basis e as the choice of the basis is arbitrary) of q, so let $\kappa_i(X) = -\ln(q_i(X))$ for $X \subseteq \Omega$. Then we define:

$$\kappa_{1,2\ldots s} = \bigcirc\!\!\!\!\cap_{i=1\ldots s}\ \kappa_i$$

So, we get

$$\kappa_{1,2\ldots s} = \sum_i \kappa_i$$

If we take $f(m_i) = \kappa_i$, and $\overline{\kappa} = \frac{1}{s}\kappa_{1,2\ldots s}$, we have a function f that satisfies the idea that the impact of the s instances in the group under consideration is equal to the impact of s times the 'average' case.

Beware that usually $\overline{\kappa}$ is not the commonality function of a belief function, even though $\bigcirc\!\!\!\!\cap_{i=1\ldots s}\overline{\kappa}$ is the commonality function of a belief function. In fact, the $\bigcirc\!\!\!\!\cap$ operator must and can innocuously be extended to 'generalized belief functions', i.e., any real function on Θ which coefficients of its Möbius transform (the equivalent of the basic belief masses) add to 1.

So our proposed intra-group distance of instances $i = 1 \dots s$ becomes:

$$D_S^2 = \frac{1}{|S|} \sum_{X \subseteq \Theta} \sum_{i \in S} (\ \kappa_i(X) - \overline{\kappa(X)}\ )^2 \tag{25}$$

where $\kappa_i(X) = -\ln q\{I_i\}(X)$ and $\overline{\kappa(X)} = \frac{1}{|S|} \sum_{i \in S} \kappa_i(X)$. The case where $q\{I_i\}(\Theta) = 0$ is solved, thanks to the continuity of all involved functions, by putting a very small mass ϵ on Θ, proceeding with the computation and taking the limit for $\epsilon \to 0$.

6 Belief decision tree parameters

In this section, we define the major parameters leading to the construction of the belief decision tree in both the averaging and the conjunctive approaches. At first, we describe what we have developed as attribute selection measures to ensure the construction of a belief decision trees. Then, we present the partitioning strategy and the stopping criteria. Finally, we detail the structure of leaves in belief decision trees.

6.1 Attribute selection measures in a belief decision tree

One of the fundamental parameters in a decision tree (and consequently in a belief decision tree) is **the attribute selection measure**. This measure is used in order to choose *"the best"* test attribute at each decision node of the tree. It quantifies the class discrimination power of each attribute.

The attribute selection measure has to provide a division of the training set into smaller subsets that are more homogeneous.

The structure of the training set is characterized by data which class is uncertain. This uncertainty is expressed by a bba on the classes domain.

6.1.1 Averaging approach

Under this approach, the attribute selection measure is based on the entropy computed from the average pignistic probability computed from the pignistic probabilities of each instance in the node. We propose the following steps to choose the appropriate attribute:

1. Compute the pignistic probability of each instance I_j in the training set by:

$$BetP^{\Theta}\{I_j\}(C_i) = \sum_{C_i \in C \subseteq \Theta} \frac{1}{|C|} \frac{m^{\Theta}\{I_j\}(C)}{1 - m^{\Theta}\{I_j\}(\emptyset)},\ \forall C_i \in \Theta \tag{26}$$

2. Compute the average pignistic probability function $BetP^{\Theta}\{S\}$ taken over the set of objects S in order to get the average probability on each class.

$$BetP^{\Theta}\{S\}(C_i) = \frac{1}{|S|} \sum_{I_j \in S} BetP^{\Theta}\{I_j\}(C_i) \tag{27}$$

3. Compute the entropy of the average pignistic probabilities in S. This value $Info(S)$ is equal to:

$$Info(S) = -\sum_{i=1}^{n} BetP^{\Theta}\{S\}(C_i).log_2 BetP^{\Theta}\{S\}(C_i) \tag{28}$$

4. Select an attribute A. Collect the subset S_v^A made with the cases having v as a value for the attribute A.

5. Compute the average pignistic probability for those cases in subset S_v^A. Let the result be denoted $BetP^{\Theta}\{S_v^A\}$ for $v \in D(A)$, $A \in \mathbf{A}$.

6. Compute $Info_A(S)$ using the same definition as suggested by Quinlan, but using the pignistic probabilities instead of the proportions. We get:

$$Info_A(S) = \sum_{v \in D(A)} \frac{|S_v^A|}{|S|} Info(S_v^A) \tag{29}$$

   where $Info(S_v^A)$ is computed from the equation (28) using $BetP^{\Theta}\{S_v^A\}$.

   The term $Info_A(S)$ is equal to the weighed sum of the different $Info(S_v^A)$ relative to the considered attribute. These $Info(S_v^A)$ are weighted by the proportion of objects in S_v^A.

7. Compute the information gain provided by the attribute A in the set of objects S such that:

$$Gain(S, A) = Info(S) - Info_A(S) \tag{30}$$

8. Using the Split Info, compute the gain ratio relative to the attribute A:

$$Gain\ Ratio(S, A) = \frac{Gain(S, A)}{Split\ Info(S, A)} \tag{31}$$

9. Repeat for every attribute $A \in \mathbf{A}$ and choose the one that maximizes the gain ratio.

Example 1: Let's continue with the previous example (see table 1 in section 4.3.4). We start by finding $Info(T)$ relative to this training set. We have to compute the average pignistic probability (see Table 4):

	C_1	C_2	C_3
$\mathbf{BetP}^{\Theta}\{T\}$	0.32	0.32	0.36

Table 4: Computation of $\mathrm{BetP}^{\Theta}\{T\}$

The results induced from the pignistic transformation mean that the average probability that a training instance chosen randomly from T belongs respectively to the classes C_1, C_2, and C_3 are respectively 0.32, 0.32 and 0.36.

These probabilities will be used to compute $Info(T)$ described as the entropy relative to the whole training set T (see equation (28)):

$$\begin{aligned} Info(T) &= -\sum_{i=1}^{3} BetP(C_i).log_2 BetP(C_i) \\ &= -0.32 * log_2 0.32 - 0.32 * log_2 0.32 - 0.36 * log_2 0.36 = 1.583. \end{aligned}$$

The value 1.583 represents the average amount of information needed to identify the class of an instance in the training set T.

Once the $Info(T)$ is calculated, we have to look for the $Info_{Hair}(T)$, $Info_{Eyes}(T)$ and $Info_{Height}(T)$. These computations will be ensured by applying the equation (29).

Let us do the computation for the eye attribute. Let $BetP^{\Theta}\{T^{Eyes}_{Blue}\}$ and $BetP^{\Theta}\{T^{Eyes}_{Brown}\}$ be the average pignistic probability functions relative respectively to the objects belonging to T and having respectively blue eyes and brown eyes (see Table 5).

	C_1	C_2	C_3
$BetP^{\Theta}\{T^{Eyes}_{Blue}\}$	0.62	0.32	0.06
$BetP^{\Theta}\{T^{Eyes}_{Brown}\}$	0.14	0.32	0.54

Table 5: Computation of $BetP^{\Theta}\{T^{Eyes}_{Blue}\}$ and $BetP^{\Theta}\{T^{Eyes}_{Brown}\}$

The next step consists on the computation of $Info_{Eyes}(T)$. By applying the equation (29), we get:

$$\begin{aligned} Info_{Eyes}(T) = &-\frac{3}{8}\sum_{i=1}^{3} BetP^{\Theta}\{T^{Eyes}_{Blue}\}(C_i).log_2 BetP^{\Theta}\{T^{Eyes}_{Blue}\}(C_i) \\ &-\frac{5}{8}.\sum_{i=1}^{3} BetP^{\Theta}\{T^{Eyes}_{Brown}\}(C_i).log_2 BetP^{\Theta}\{T^{Eyes}_{Brown}\}(C_i) \\ = &\ 1.299 \end{aligned}$$

Then, we compute the information gain (see equation 30), we get respectively:

$$\begin{aligned}
Gain(T, Eyes) &= Info(T) - Info_{Eyes}(T) \\
&= 1.583 - 1.299 = 0.284, \\
Split\ Info(T, Eyes) &= 0.95, \\
Gain\ ratio(T, Eyes) &= 0.3.
\end{aligned}$$

By applying the same process, we get

$$\begin{aligned}
Gain\ Ratio(T, Hair) &= 0.35, \\
Gain\ Ratio(T, Height) &= 0.11.
\end{aligned}$$

The attribute that maximizes the gain ratio is the hair attribute. It will be chosen as the root of the decision tree relative to the training set T and branches are created for each of its possible values (Blond, Red, Dark). The same procedure is applied, iteratively, to the cases that fall in each subset that is created according to the values of the selected attribute.

6.1.2 Conjunctive approach

The conjunctive approach uses the intra-group distance D_S^2 (see equation (25)) that quantifies for each attribute value how much objects are close from each others.

The selection attribute measure to build a belief decision tree under the conjunctive approach is made of the following steps:

1. For each case in the training set, compute :

$$\kappa\{I_j\}(C) = -\ln q^{\Theta}\{I_j\}(C)\ \forall C \subseteq \Theta \tag{32}$$

   from the bba $m^{\Theta}\{I_j\}$.

2. For each attribute value v of attribute A, compute the joint $\kappa\{S_v^A\}$ defined on Θ by:

$$\kappa\{S_v^A\} = \sum_{I_j \in S_v^A} \kappa\{I_j\} \tag{33}$$

3. Hence for each attribute value, the intra-group distance $SumD(S_v^A)$ is defined by:

$$SumD(S_v^A) = \frac{1}{|S_v^A|} \sum_{I_j \in S_v^A} \sum_{X \subseteq \Theta} (\kappa\{I_i\}(X) - \frac{1}{|S_v^A|}\kappa\{S_v^A\}(X))^2 \tag{34}$$

4. Once the different $SumD(S_v^A)$ are calculated, for each attribute $A \in \mathbf{A}$, compute $SumD_A(S)$ representing the weighted sum of the different $SumD(S_v^A)$ relative to each value v of the attribute A:

$$SumD_A(S) = \sum_{v \in D(A)} \frac{|S_v^A|}{|S|} SumD(S_v^A) \tag{35}$$

5. At this level, we may conclude which attribute will be chosen as a root relative to the set of objects S. It consists in selecting the one presenting the minimal $SumD_A(S)$. In other words the attribute presenting a partition of objects in which objects are the closest from each others. Nevertheless we can proceed in order to take into account the number of possible values for the domain of the attribute.

6. By analogy to our averaging approach, we may also compute $Diff(S, A)$ defined as the difference between $SumD(S)$ and $SumD_A(S)$:

$$Diff(S, A) = SumD(S) - SumD_A(S) \tag{36}$$

where

$$SumD(S) = \frac{1}{|S|} \sum_{I_j \in S} \sum_{X \subseteq \Theta} (\kappa\{I_i\}(X) - \frac{1}{|S|}\kappa\{S\}(X))^2 \tag{37}$$

7. Using the Split Info, compute the Diff ratio relative to the attribute A:

$$Diff\ Ratio(S, A) = \frac{Diff(S, A)}{Split\ Info(S, A)} \tag{38}$$

8. Repeat for every attribute $A \in \mathbf{A}$ and choose the one that maximizes the gain ratio.

Example 2: We use the training set presented in the previous example and apply the attribute selection measure based on a conjunctive approach in order to find the test attribute.

We start by computing $SumD(T)$, the sum of distances separating each training instance to the whole set T. We have:

$$\begin{aligned} m\{T\} &= m^{\Theta}\{I_1\} \circledcirc m^{\Theta}\{I_2\} \circledcirc ... \circledcirc m^{\Theta}\{I_8\} \\ SumD(T) &= \frac{1}{8} \sum_{I_j \in T} D(I_j, T) \\ &= \frac{1}{8}(D(I_1, T) + D(I_2, T) + ... + D(I_8, T)) \\ &= 3.14; \end{aligned}$$

We proceed with the computation for the eye attribute. Two values may be possible which are blue eyes or brown eyes. We define the following bba's:

$$m\{T_{Blue}^{Eyes}\} = m^{\Theta}\{I_1\} \circledcirc m^{\Theta}\{I_3\} \circledcirc m^{\Theta}\{I_5\}$$

and $m\{T_{Brown}^{Eyes}\} = m^{\Theta}\{I_2\} \circledcirc m^{\Theta}\{I_4\} \circledcirc m^{\Theta}\{I_6\} \circledcirc m^{\Theta}\{I_7\} \circledcirc m^{\Theta}\{I_8\}$

$$SumD_{Eyes}(T) = \frac{|T_{Blue}^{Eyes}|}{|T|} SumD_(T_{Blue}^{Eyes}) + \frac{|T_{Brown}^{Eyes}|}{|T|} SumD_(T_{Blue}^{Eyes}) \quad = 2.13$$

By applying the same process for the other attributes, we get:

$$SumD_{Hair}(T) = 1.87,$$
$$SumD_{Height}(T) = 2.79$$

We have also to compute $Diff(T, A)$ for $A \in \{Hair, Eyes, Height\}$. We get:

$$Diff(T, Hair) = 1.27$$
$$Diff(T, Eyes) = 1.01$$
$$Diff(T, Height) = 0.34$$

Then, the computation of the Diff ratio gives as the following results:

$$Diff\ Ratio(T, Hair) = 0.91$$
$$Diff\ Ratio(T, Eyes) = 1.06$$
$$Diff\ Ratio(T, Height) = 0.43.$$

The application of the "*Diff Ratio*" criterion leads to the choice of the eye attribute as an attribute test relative to the training set T.

6.2 Partitioning strategy

The partitioning strategy, also known as the splitting strategy, defines how to split the training set according to the attribute values. Since we deal with symbolic attributes, we create an edge for each value of the attribute chosen as a decision node. Thus, we get several training subsets where each one is relative to one branch and regrouping objects having the same attribute value.

The partitioning strategy for the construction of a belief decision tree is very similar to the partitioning strategy used in the classic tree. This is due to that the uncertainty concerns the classes of the training instances and not the values of their attributes.

6.3 Stopping criteria

The stopping criteria control the process of the construction of the belief decision tree. It allows to stop the development of a path and to declare the node as a leaf. In other words, it determines whether or not a training subset should be further divided.

Three strategies are proposed as a stopping criterion:

1. If the treated node includes only one instance. Hence, the leaf will contain only one object.

2. If the treated node includes instances of which the m_i's are equal.

3. If there is no further attribute to test. In other words, if all the attributes are split.

4. If the value of the applied attribute selection measure for the remaining attributes is less or equal than zero which means that the *"eventual"* partition does not provide a better separation.

In such case, a leaf will include one or several instances characterized by the same values for the selected attributes but generally having different bba's on their actual classes.

6.4 Structure of leaves

The leaf in a belief decision tree will be labeled by a basic belief assignment function since the classes of the different training instances are expressed by the means of basic belief assignments.

The major question is how to compute each leaf's bba? In fact, two cases must be treated:

6.4.1 In the averaging approach

Using the averaging approach in the selection attribute measure, the leaf's bba will be defined as following:

1. When only one object belongs to the leaf S, the leaf's bba would be equal to this object's bba as defined in the training set.

2. When there are many objects attached to the leaf S, the leaf's bba would be equal to the average of the different basic belief assignment functions relative to these objects.

$$m^{\Theta}\{S\}(C) = \frac{1}{|S|} \sum_{I_j \in S} m^{\Theta}\{I_j\}(C) \tag{39}$$

6.4.2 In the conjunctive approach

Using the conjunctive approach for the development of the selection attribute measure, the leaf's bba will be defined as following:

1. When only one object belongs to the leaf S, the leaf's bba would be equal to this object's bba defined on the training set.

2. When there are many objects attached to the leaf S, the leaf's bba would be equal the result of the conjunctive combination of these objects' bba by using the conjunctive rule.

$$m^{\Theta}\{S\} = \textcircled{\cap}_{I_j \in S}\; m\{I_j\} \tag{40}$$

7 Building procedure

The building procedure is also called **the induction task**. It allows to induce a "belief" decision tree in order to use it for the classification task.

Even in an uncertain context, the building of belief decision trees requires a top down approach based on the conquer and divide principle.

7.1 Description

As mentioned, the algorithm to construct a decision tree is based on three major parameters: the attribute selection measure, the partitioning strategy, the stopping criteria. These parameters must take into account the uncertainty encountered in the training set.

In fact, a belief decision tree is constructed from a training set of objects based on successive refinements. These refinements based on both the attribute selection measure and a partitioning strategy lead to small training subsets. The process may be repeated until leaves are encountered. Such nodes have to satisfy the stopping criteria.

7.2 Algorithm of building a belief decision tree

Let T be a training set composed by objects characterized by l symbolic attributes $(A_1, A_2, ..., A_l)$ and that may belong to the set of classes $\Theta = \{C_1, C_2, ...,$ $C_n\}$. For each object $I_j (j = 1 \dots p)$ of the training set will correspond a basic belief assignment $m^\Theta\{I_j\}$ expressing the quantity of beliefs exactly committed to the subsets of classes.

Our algorithm which uses a Top Down Induction of Decision Trees (TDIDT) approach, will have the same skeleton as an ID3 algorithm [20]. Besides, our algorithm is considered as generic since it provides two possibilities for selecting the attributes by using either the averaging approach or the conjunctive one.

The different steps of our algorithm of building a belief decision tree are described as follows:

1. Generate the root node of the decision tree including all the objects of the training set T.

2. Choose which approaches will be applied : either the averaging approach or the conjunctive one.

3. Verify if this node satisfies or not the stopping criteria (see section 6.3) :

   - If yes, declare it as a leaf node and compute its corresponding bba according to the chosen approach (see section 6.4).
   - If not, look for the attribute having the highest attribute selection measure (see section 6.1). This attribute will be designed as the root of the decision tree related to the whole training set.

4. Apply the partitioning strategy (see section 6.2) by developing an edge for each attribute value chosen as a root. This partition leads to several training subsets.

5. Create a root node relative to each training subset.

6. Repeat the same process for each training subset from the step 3, while verifying the stopping criterion.

7. Stop when all the nodes of the latter level of the tree are leaves.

Example 3: Let's continue with the example proposed in section 4.3.4. Let's generate the belief decision trees relative respectively to the average approach (the one relative to the conjunctive approach may be done in the same manner).

As computed in the example 1, we have found that:

$$\begin{aligned} Gain\ Ratio(T, Hair) &= 0.35, \\ Gain\ Ratio(T, Eyes) &= 0.3, \\ Gain\ Ratio(T, Height) &= 0.11. \end{aligned}$$

Neither of these attributes satisfy the stopping criteria, so we choose the hair attribute as the root of the decision tree relative to the training set T, since it presents the highest gain ratio.

Therefore, branches are created below this root for each of its possible value (Blond, Red, Dark).

We get the following belief decision tree (see figure 1):

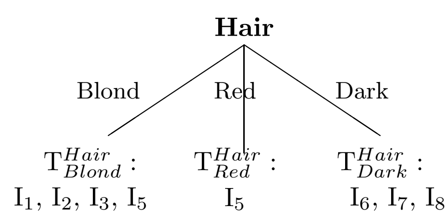

Figure 1: First generated belief decision tree

We notice that the training subset T^{Eyes}_{Red} contains only one object. Hence, the stopping criteria are fulfilled for this subset. As a consequence, the node relative to T^{Eyes}_{Red} is declared as a leaf and its corresponding bba will be equal to $m^{\Theta}\{I_5\}$.

For the training subsets T^{Eyes}_{Blond} and T^{Eyes}_{Dark}, we apply the same process as we did for the training set T until the stopping criteria hold.

The final belief decision tree induced by our algorithm is given by (see figure 2):

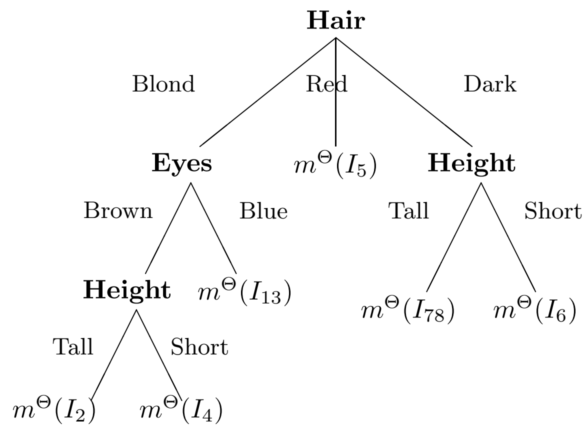

Figure 2: Belief decision tree representing training scenarios.

where $m^{\Theta}\{I_{13}\}$ and $m^{\Theta}\{I_{78}\}$ are respectively the average bba's relative to the objects $\{I_1\}$ and $\{I_3\}$ for $\mathrm{m}^{\Theta}\{I_{13}\}$ and to the objects $\{I_7\}$ and $\{I_8\}$ for $\mathrm{m}^{\Theta}\{I_{78}\}$.

So we get (see equation 39):
$m^{\Theta}\{I_{13}\}(C_1) = 0.85$; $m^{\Theta}\{I_1\}(\Theta) = 0.15$;
$m^{\Theta}\{I_{78}\}(C_3) = 0.7$; $m^{\Theta}\{I_{78}\}(C_1 \cup C_3) = 0.1$; $m^{\Theta}\{I_{78}\}(\Theta) = 0.2$;

8 Classification procedure

Once the belief decision tree is constructed, the following procedure will be the classification of new instances referring to as new objects. Such task is also named **the inference task**.

As we deal with an uncertain environment, several cases regarding the knowledge of the attribute values have been studied in order to ensure classification using a belief decision tree.

8.1 Standard classification

Our method is able to ensure the standard classification where each attribute value (of the new instance to classify) is assumed to be exact and certain.

As in an ordinary tree, it consists in starting from the root node and repeating to test the attribute at each node by taking into account the attribute value until reaching a leaf.

Contrary to the classical decision tree where a unique class is attached to the leaf, in our belief decision tree the new instance class will be defined by a basic belief assignment related to the reached leaf. This bba defined on the set of classes, represents beliefs on the different subsets of classes (singletons, disjunctions) of the new instance to classify.

In order to make a decision and to get the probability of each singular class, we propose to apply the pignistic transformation.

8.2 Disjunctive case

We consider now the classification of new instances poised by uncertainty in the values of their attributes.

Suppose that the value of an attribute of the new object to classify is not precisely known, and it is only known to belong to a set of possible values of in the domain of this attribute. In particular, when the value is missing, the set is equated to the whole domain of the attribute. This can even occur for several attributes. We assume that the various intervals are non interactive, i.e., if all we know is that the value of attribute A_1 is in $\theta_1 \subseteq \Theta$ and that the value of attribute A_2 is in $\theta_2 \subseteq \Theta$, then all the values in $\theta_1 \times \theta_2$ are possible. Nevertheless the algorithm could easily be adapted to the interactive case.

In order to classify such an object, we determine all the leaves which the object could belong to by tracing out all the paths compatible with our knowledge about the different attribute values.

As a consequence, the new instance may belong to many leaves. In each one, we have a bba representing our knowledge about the class to which belong the cases in the leaf. These bba's must be combined in order to get the belief on the instance's class. The disjunctive rule of combination developed by Smets (see equation 18) is the appropriate operator to combine these bba's as it produces the bba under the hypothesis that one path is true (but we do not know which path).

When a decision has to be made, the bba induced from the disjunctive rule is transformed into a probability function by applying the pignistic transformation, producing the probabilities that the new instance belongs to this or that class.

Note that as we mentioned, this case includes the total ignorance of some attribute values. When we deal with unknown attribute values, all the branches relative to the considered attribute will be taken into account. Then, the same process (as in the case of the disjunctive values) will be applied.

8.3 General case

The uncertainty characterizing the new instance to classify is not necessarily represented by disjunctive values or missing values (total ignorance). It may be more complicated, especially when the attribute values are provided by several experts. Therefore, it would be interesting to extend our classification procedure in order to handle more general uncertainty.

The uncertainty about the value of the attribute can be defined by a bba on the set of all the possible values of the attribute. Such bba's may be given by one expert or result from the combination (using the conjunctive rule) of several bba's on the attribute values.

Let:

- m^{A_i} be the bba representing the parts of beliefs committed exactly to the different values relative to the attribute A_i of the new instance to classify. This bba is defined on the frame of discernment Θ_{A_i} including all the possible values of the attribute A_i.

- Θ_A be the global frame of discernment relative to all the attributes. It is equal to the cross product of the different Θ_{A_i}. We denote by:

$$\Theta_A = \times_{i=1..k}\Theta_{A_i} \tag{41}$$

Since an instance is characterized by a set of combination of values where each one is relative to an attribute, we have firstly to find the bba expressing beliefs on both the different attributes' values of the new instance to classify. In other words, we have to look for the joint bba representing beliefs on all the instance's attributes. To ensure this objective, we have to apply the following steps:

1. Extend the different bba's m^{A_i} to the global frame of attributes Θ_A. As a result, we get the different bba's $m^{A_i \uparrow A}$.

2. Combine the different extended bba's by applying the conjunctive rule of combination.

$$m^{\Theta_A} = \circledcirc_{i=1..k} m^{A_i \uparrow A} \tag{42}$$

Thus, we get a joint bba representing beliefs on the different combinations of the attributes characterizing the given instance. Should there be some 'correlation' between the bba's, the procedure could be adapted, but in any case the end product is a bba on Θ_A.

We then, consider individually the focal elements of the bba m^{Θ_A}. Let x be such a focal element. The next step in our classification task is to compute the belief functions $bel^{\Theta}[x]$. The computation of this function depends on the subset x and more exactly on the focal elements of the bba m^{Θ_A}:

1. If the treated focal element x is a singleton (only one value for each attribute), then $bel^{\Theta}[x]$ is equal to the belief function corresponding to the leaf to which this focal element is attached.

2. If the focal element x is not a singleton (some attributes have more than one value), that is it contains a disjunction in some attribute values. Then, we have to explore all the possible paths relative to this combination of values. Two cases are possible:

   - If these paths lead to one leaf, then $bel^{\Theta}[x]$ is equal to this leaf's belief function.

- If these paths lead to distinct leaves, then $bel^{\Theta}[x]$ is equal to the result of the disjunctive combination of each leaf's belief function by applying the disjunctive rule.

Finally the belief functions computed with each focal element x are averaged using the m^{Θ_A}:

$$bel^{\Theta}[m^{\Theta_A}](\theta) = \sum_{x \subseteq \Theta_A} m^{\Theta_A}(x).bel^{\Theta}[x](\theta) \text{ for } \theta \in \Theta \tag{43}$$

$bel^{\Theta}[m^{\Theta_A}]$ gives us total beliefs on the classes (subsets of classes) that the new instance may to belong. You may also compute the corresponding basic belief assignment $m^{\Theta}[m^{\Theta_A}]$.

To make a decision, this belief function (or bba) is transformed to a probability function on singular classes via the pignistic transformation.

Example 4: In order to illustrate our generalized case, we consider the instance to classify characterized by:

- Eyes = Blue which is equivalent to a certain bba m^{Eyes} having only the value blue as a focal element: $m^{Eyes}(\{Blue\}) = 1$;
- Height = Tall which is equivalent to a certain bba m^{Height} having only the value tall as a focal element: $m^{Height}(\{Tall\}) = 1$;
- However, the value relative to the hair attribute is uncertain and described by the bba m^{Hair} such that: $m^{Hair}(\{Dark\}) = 0.6$; $m^{Hair}(\{Dark \cup Red\}) = 0.2$; $m^{Hair}(\Theta_{Hair}) = 0.2$;

Let $\Theta_A = \Theta_{Hair} \times \Theta_{Eyes} \times \Theta_{Height}$

So, Θ_A = {(Blond, Blue, Tall), (Blond, Blue, Short), (Blond, Brown, Tall), (Blond, Brown, Short), (Red, Blue, Tall), (Red, Blue, Short), (Red, Brown, Tall), (Red, Brown, Short), (Dark, Blue, Tall), (Dark, Blue, Short), (Dark, Brown, Tall), (Dark, Brown, Short) }

The extension of the different bba's to Θ_A gives as a result:

- the bba m^{Hair} induces the following bba on Θ_A:

$$m^{Hair\uparrow\Theta_A}(\{Dark\} \times \Theta_{Eyes} \times \Theta_{Height}) = 0.6$$
$$m^{Hair\uparrow\Theta_A}(\{Dark \cup Red\} \times \Theta_{Eyes} \times \Theta_{Height}) = 0.2$$
$$m^{Hair\uparrow\Theta_A}(\Theta_{Hair} \times \Theta_{Eyes} \times \Theta_{Height}) = 0.2.$$

- the bba m^{Eyes} induces the following bba on Θ_A:

$$m^{Eyes\uparrow\Theta_A}(\Theta_{Hair} \times \{Blue\} \times \Theta_{Height}) = 1.$$

- the bba m^{Height} induces the following bba on Θ_A:

$$m^{Height\uparrow\Theta_A}(\Theta_{Hair} \times \Theta_{Eyes} \times \{Tall\}) = 1.$$

Once the attributes' bba's are extended to Θ_A, then we can apply the conjunctive rule. The result of this combination will be a joint bba on singular instances or subsets of instances. So we get:

$$m^{\Theta_A} = m^{Hair\uparrow\Theta_A} \circledcirc m^{Eyes\uparrow\Theta_A} \circledcirc m^{Height\uparrow\Theta_A}$$

where:

$$\begin{aligned} m^{\Theta_A}(Dark \times Blue \times Tall) &= 0.6, \\ m^{\Theta_A}(\{Dark \cup Red\} \times \{Blue\} \times \{Tall\}) &= 0.2, \\ m^{\Theta_A}(\Theta_{Hair} \times \{Blue\} \times \{Tall\}) &= 0.2. \end{aligned}$$

Next, we have to find beliefs on classes (defined on Θ) given the values of the attributes characterizing the new instance to classify. In fact, three belief functions have to be defined where for each one, we take into account one focal element of m^{Θ_A}. According to the belief decision tree generated (see figure 2), we get:

$$\begin{aligned} bel^{\Theta}[\{(Dark, Blue, Tall)\}] &= bel_{78}; \\ bel^{\Theta}[\{Dark \cup Red\} \times \{Blue\} \times \{Tall\}] &= bel_5 \circledcirc bel_{78}; \\ bel^{\Theta}[\Theta_{Hair} \times \{Blue\} \times \{Tall\}] &= bel_{13} \circledcirc bel_5 \circledcirc bel_{78} \end{aligned}$$

	C_1	C_2	C_3	$C_1 \cup C_2$	$C_1 \cup C_3$	$C_2 \cup C_3$	Θ
bel_{78}	0	0	0.7	0	0.8	0.7	1
$bel_5 \circledcirc bel_{78}$	0	0	0	0	0	0.56	1
$bel_{13} \circledcirc bel_5 \circledcirc bel_{78}$	0	0	0	0	0	0	1

Table 6: Beliefs on classes given the attributes' values

Hence, these belief functions will be averaged (see equation (42)), we get:

$$\begin{aligned} bel^{\Theta}[m^{\Theta_A}](C_1) &= 0.6 * 0 + 0.2 * 0 + 0.2 * 0 = 0; \\ bel^{\Theta}[m^{\Theta_A}](C_2) &= 0; \\ bel^{\Theta}[m^{\Theta_A}](C_3) &= 0.6 * 0.7 + 0.2 * 0 + 0 * 0 = 0.42; \\ bel^{\Theta}[m^{\Theta_A}](C_1 \cup C_2) &= 0; \\ bel^{\Theta}[m^{\Theta_A}](C_1 \cup C_3) &= 0.6 * 0.8 + 0.2 * 0 + 0.2 * 0 = 0.48; \\ bel^{\Theta}[m^{\Theta_A}](C_2 \cup C_3) &= 0.6 * 0.7 + 0.2 * 0.56 + 0.2 * 0 = 0.532. \end{aligned}$$

Hence, we get:

$$m^{\Theta}[m^{\Theta_A}](C_1) = 0; m^{\Theta}[m^{\Theta_A}](C_2) = 0; m^{\Theta}[m^{\Theta_A}](C_3) = 0.42;$$
$$m^{\Theta}[m^{\Theta_A}](C_1 \cup C_2) = 0; m^{\Theta}[m^{\Theta_A}](C_1 \cup C_3) = 0.06;$$
$$m^{\Theta}[m^{\Theta_A}](C_2 \cup C_3) = 0.112; m^{\Theta}[m^{\Theta_A}](\Theta) = 0.408;$$

Each belief mass represents the part of belief allocated to the fact that the given instance belong to the focal element of this mass.

Applying the pignistic transformation, the pignistic probability will be defined as follows:

$$BetP(C_1) = 0.166; BetP(C_2) = 0.192; BetP(C_3) = 0.642.$$

We notice that the probability that this instance belongs respectively to the classes C_1, C_2 and C_3 are respectively 0.166, 0.192 and 0.642.

As a consequence, it is most probable that this new instance (characterized by blue eyes, tall height and uncertain in the color of its hair) to belong to the class C_3.

9 Conclusion

In this paper, we have defined the belief decision tree which is a new technique associating the decision tree method to the belief function theory in order to handle uncertainty that can exist on classification problem parameters.

We consider the case where the knowledge about the class of the instances in the training set is represented by a belief function over the set of possible classes.

We present two attribute selection measures using the belief function formalism, one parallel to Quinlan's measure based on the entropy (the averaging method), the other close in spirit to the TBM (the conjunctive method). Partitioning strategy and stopping criteria are provided, and the meaning of the data in the leaves is detailed. We present the different steps of the procedure allowing the construction of the tree in an uncertain context.

Next, we present the inference task ensuring the classification of new instances using the constructed belief decision tree. We consider the case where the knowledge about the value of some attributes is represented by a bba, and show how to use the belief decision tree. All the leaves compatible with the knowledge are considered and their individual conclusions are combined by the disjunctive rule of combination. Classification is then based on the pignistic probabilities derived from the global bba.

This paper presents the theoretical concepts underlying the belief decision trees. The evaluation of the belief decision tree and the comparison of its results with those obtained by classical methods will be reported in a forthcoming paper.

References

[1] J. A. Barnett, Combining Opinions About the Order of Rule Execution, The Ninth National Conference on Artificial Intelligence, AAAI 91, pp 477-481, 1991.

[2] M. S. Bjanger, Induction of decision trees from partially classified data using belief functions, Master, University of Compiegne, 2000.

[3] L. Breiman, J. H. Friedman, R. A. Olshen, C. J. Stone, Classification and regression trees, Monterey, CA : Wadsworth & Brooks, 1984.

[4] J. Cheng, U. M. Fayyad, K. B. Irani, Z. Qian, Improved decision trees : a generalized version of ID3, Proceedings of the fifth Internatioanl Conference on Machine Learning, pp 100-106, June 12-14, 1988.

[5] T. Denoeux, A k-nearest neighbor classification rule based on Dempster-Shafer theory, IEEE Transactions on Systems, Man, and Cybernatics, Vol 25, N5, pp 804-813, May, 1995.

[6] T. Denoeux, Reasoning with imprecise belief structures, International Journal of Approximate Reasoning 20, pp 79-111, 1999.

[7] T. Denoeux, M. S. Bjanger, Induction of decision trees from partially classified data using belief functions, To appear in the proceeding of SMC Conference, 2000.

[8] Z. Elouedi, K. Mellouli, P. Smets, Decision trees using the belief function theory the proceedings of the International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems IPMU'2000, pp. 141-148, 2000.

[9] Z. Elouedi, K. Mellouli, P. Smets, Classification with belief decision trees To appear in the proceedings of the The Ninth International Conference on Artificial Intelligence: Methodology, Systems, Applications, AIMSA'2000, 2000

[10] U. M. Fayyad, K. B. Irani, The attribute selection problem in decision tree generation, The Tenth National Conference on Artificial Intelligence, AAAI 92, pp 104-110, 1992.

[11] J. Gordon, E. H. Shortliffe, A method for managing evidential reasoning in a hierarchical hypothesis space, Aritificial intelligence, Vol 26, pp 323-357, 1985.

[12] F. Janez, Fusion de sources d'information définies sur des référentiels non exhaustifs différents Thèse de doctorat, Ecole Doctorale Sciences pour l'Ingénieur de Nantes, 1996.

[13] G. Z. Janikow, Fuzzy decision trees : Issues and methods, IEEE Transactions on Systems, Man and Cybernatics- Part B : Cybernatics, Vol 28 N1, pp 1-14, February, 1998.

[14] R. Kennes, Evidential reasoning in a categorical perspective: conjunction and disjunction of belief functions, Technical Report No. TR/IRIDIA/91-6, Université Libre de Bruxelles, 1991.

[15] J. Kholas, P. A. Monney, A mathematical theory of hints. An approach to Dempster-Shafer theory of evidence, Lecture notes on Economics and Mathematical systems No.425. Springer-Verlag, 1995.

[16] W. Z. Liu, A. P. White, The importance of attribute selection measures in decsion tree induction, Machine Learning 15, pp 24-41, 1994.

[17] R. Lopez De Mantaras, Approximate reasoning models, Published by Ellis Horwood, England, 1990.

[18] C. Marsala, Apprentissage inductif en présence de données imprécises : Construction et utilisation d'arbres de décision flous, Thèse de doctorat de l'Université Paris6, LIP6, 1998.

[19] K. Mellouli, On the propagation of beliefs in networks using the Dempster-Shafer theory of evidence, Ph.D dissertation, School of business, University of Kansas, Lawrence, KS, 1987.

[20] J. R. Quinlan, Induction of decision trees, Machine Learning 1, pp 81-106, 1986.

[21] J. R. Quinlan, Decision trees as probabilistic classifiers, Proceedings of the Fourth International Workshop on Machine Learning, pp 31-37, June 22-25, 1987.

[22] J. R. Quinlan, Decision trees and decision making, IEEE Transactions on Systems, Man and Cybernatics- Vol 20 N2, pp 339-346 March/April, 1990.

[23] J. R. Quinlan, Probabilistic decision trees, In R. S. Michalski, J. G. Carbonell, T. M. Michell, editors, Machnie learning, Volume 3, Chapitre 5, pp 140-152, Morgan Kaufmann Publishers, 1990.

[24] J. R. Quinlan, C4.5 : Programs for machine learning, Morgan Kaufmann, San Mateo, Ca, 1993.

[25] S. R. Safavian, D. Landgrebe, A survey of decision tree classifier methodology, IEEE transactions on systems, man and cybernatics, vol 21 N3, pp 660-674, May/June, 1991.

[26] G. Shafer, A mathematical theory of evidence, Princeton University Press, Princeton NJ, 1976.

[27] G. Shafer, Perspectives on the theory and practice of belief functions, International Journal of Approximate Reasoning, Vol 4, pp 323-362, 1990.

[28] G. Shafer, Rejoinders to comments on perspectives on the theory and practice of belief functions, International Journal of Approximate Reasoning, Vol 6, pp 445-480, 1992.

[29] C. E. Shannon, A mathematical theory of communication, The Bell system technical journal, Vol VII, N3, pp 379-423, July, 1948.

[30] P. P. Shenoy, G. Shafer, Axioms for probability and belief-function propagation, Edited by Shafer G. and Pearl J. (The Morgan Kauffman series in representation and reasonning), pp 574-610, 1990.

[31] P. Smets, Decisions and belief functions, Technical Report No TR/IRIDIA/90-10, Université Libre de Bruxelles, 1990.

[32] P. Smets, The combination of evidence in the Transferable Belief Model, IEEE Transactions on Pattern Analysis and Machine Intelligence 12, pp 321-344, 1990.

[33] P. Smets, Belief functions, Non Standard Logics for Automated Reasoning. Smets Ph., Mamdani A., Dubois D. and Prade H. (Editors) Academic Press, London pp 253-286, 1990.

[34] P. Smets, Belief functions : the disjunctive rule of combination and the generalized bayesian theorem, International Journal of Approximate Reasoning 9, pp 1-35, 1993.

[35] P. Smets, R. Kennes, The transferable belief model, Artificial Intelligence, Vol 66, pp 191-234, 1994.

[36] P. Smets, Theories of uncertainty, Technical Report No. TR/IRIDIA/95-20, Université Libre de Bruxelles, 1995.

[37] P. Smets, Numerical representation of uncertainty, D. M. Gabbay and Ph. Smets (eds.), Handbook of Defeasible Reasoning and Uncertainty Management Systems, Vol. 3, pp 265-309, Klower Academic Publishers, 1998.

[38] P. Smets, The Transferable Belief Model for Quantified Belief Representation, D.M. Gabbay and Ph. Smets (eds.), Handbook of Defeasible Reasoning and Uncertainty Management Systems, Vol. 1, pp 267-301, Kluwer, Doordrecht, 1998.

[39] P. Smets, The Application of the Transferable Belief Model to Diagnostic Problems, Int. J. Intelligent Systems, 13, pp 127-158, 1998.

[40] P. Smets, Practical Uses of Belief Functions, Laskey K. B. and Prade H. (eds.) Uncertainty in Artificial Intelligence 15. UAI99 pp 612-621, 1999

[41] M. Umano, H. Okamoto, I. H. H. Tamura, F. Kawachi, S. Umedzu, J. Kinoshita, Fuzzy decision trees by fuzzy ID3 algorithm and its application to diagnosis systems, Proceedings of third IEEE International Conference on Fuzzy Systems, FUZZ-IEEE'94, pp 2113-2118, June 26-29, 1994.

[42] Paul E. Utgoff, Incremental induction of decision trees, Machine Learning, 4, pp 161-186, 1989.

[43] Y. Yuan, M. J. Shaw, Induction of fuzzy decision trees, Fuzzy sets and Systems 69, pp 125-139, 1995.

[44] J. Zeidler, M. Schlosser, Continuous valued attributes in fuzzy decision trees, IPMU 96, pp 395-400, 1996.