

Decision trees using the belief function theory

Zied Elouedi
LAROID,
Institut Supérieur de Gestion
de Tunis,
41 rue de la liberté, Le Bardo
2000, Tunisie.
zied.elouedi@isg.rnu.tn

Khaled Mellouli
LAROID,
Institut Supérieur de Gestion
de Tunis,
41 rue de la liberté, Le Bardo
2000 Tunisie.
khaled.mellouli@ihc.rnu.tn

Philippe Smets
IRIDIA,
Université Libre de Bruxelles,
50 av., F.Roosevelt, CP19416,
1050 Bruxelles, Belgique.
psmets@ulb.ac.be

Abstract

This paper presents an algorithm for building decision trees in an uncertain environment. Our algorithm will use the theory of belief functions in order to represent the uncertainty about the parameters of the classification problem. Our method will be concerned with both the decision tree building task and the classification task.

Keywords: Belief function theory, Decision tree, Classification.

1 Introduction

Decision trees are one of the most widely used classification techniques especially in artificial intelligence. Their popularity is basically due to their ability to express knowledge in a formalism that is often easier to interpret by experts and even by ordinary users.

Despite their accuracy when precise and certain data are available, the classical versions of decision tree algorithms are not able to handle the uncertainty in classification problems. Hence, their results are categorical and do not convey the uncertainty that may occur in the attribute values or in the case class.

To overcome this limitation, Quinlan has developed probabilistic decision trees [5] where his major objective is to deal with examples characterized by missing or imprecise attribute values. However within his framework, only statistical uncertainty induced by information arisen from random behavior, is taken into account.

In this paper, we present a classification method based on the decision tree approach having the objective to cope with the uncertainty that may occur in a classification problem and which is basically related to human thinking, reasoning and cognition.

Our algorithm will use the belief function theory as understood in the transferable belief model (TBM) [11, 12] and which seems offering a convenient framework thanks to its ability to represent epistemological uncertainty.

Moreover, the TBM allows experts to express partial beliefs in a much more flexible way than probability functions do. It also allows to handle partial or even total ignorance concerning classification parameters. In addition to these advantages, it offers appropriate tools to combine several pieces of evidence.

This paper is composed as follows: we start by introducing decision trees, then we give an overview of the basic concepts of the belief function theory. In the main part of the paper, we present our decision tree algorithm based on the evidence theory. The two major phases will be detailed: the building of a decision tree and the classification task. Our algorithm will be illustrated by an example in order to understand its real unfolding.

2 Decision trees

Decision trees present a system using a top-down strategy based on the divide and conquer approach where the major aim is to partition the tree in many subsets mutually exclusive. Each subset of the partition represents a classification sub problem.

A decision tree is a representation of a decision procedure allowing to determine the class of a case. It is composed of three basic elements [14]:

- Decision nodes specifying the test attributes.
- Edges corresponding to the possible attribute outcomes.
- Leaves named also answer nodes and labeled by a class.

The decision tree classifier is used in two different contexts:

1. building decision trees where the main objective is to find at each decision node of the tree, the best test attribute that diminishes, as much as possible, the mixture of classes with each subset created by the test.
2. classification where we start by the root of the decision tree, then we test the attribute specified by this node. The result of this test allows to move down the tree branch relative to the attribute value of the given example. This process will be repeated until a leaf is encountered. So, the case is classified by tracing out a path from the root of the decision tree to one of its leaves [6].

Many algorithms have been developed for learning decision trees. Among the most famous are those suggested by Quinlan like ID3 [4] and C4.5 [7]. One of the fundamental steps of these algorithms is the selection measure applied to find the appropriate test attribute at each decision node while growing the tree.

Quinlan has defined a measure called information gain [4] also referred to as the criterion gain originally based on the information theory of Shannon. 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 A be an attribute with k outcomes that partitions the training set T of instances into k subsets T_m ($m = 1..k$).

Suppose there are n classes, denoted C_1, \dots, C_n , and $p_i = \frac{\text{freq}(C_i, T)}{|T|}$ represents the proportion of objects

in T belonging to the class C_i ($i = 1..n$).

The information gain for an attribute A relative to the training set T is defined as following [7]:

$$\text{Gain}(T, A) = \text{Info}(T) - \text{Info}_A(T)$$

$$\text{where Info}(T) = - \sum_{i=1}^n p_i \cdot \log_2 p_i$$

$$= - \sum_{i=1}^n \frac{\text{freq}(C_i, T)}{|T|} \cdot \log_2 \left(\frac{\text{freq}(C_i, T)}{|T|} \right)$$

$$\text{and Info}_A(T) = \sum_{m=1}^k \frac{|T_m|}{|T|} \cdot \text{Info}(T_m)$$

$\text{Info}(T)$ measures the average amount of information needed to identify the class of a case in a training set T, it is also called the entropy of the set T [7]. $\text{Info}(T_m)$ is the same measure but computed from the data in T_m . $\text{Info}_A(T)$ is the weighted average of the conditional entropies. It represents the same type of measurement as $\text{Info}(T)$ but after considering the partition of T obtained by taking into account the k outcomes of the attribute A. This attribute selection criterion computes the difference between the entropies before and after the partition, the largest difference corresponds to the best attribute.

3. Belief function theory

In this section, the main concepts of the belief function model are recalled [8, 11, 12].

3.1 Background

Let Θ be the frame of discernment representing a finite set of elementary hypotheses related to a problem domain. We denote by 2^Θ the set of all the subsets of Θ .

To represent degrees of belief, Shafer [8] introduces the so-called basic belief assignments (called initially basic 'probability' assignments, an expression that has created serious confusion). They quantify the part of belief that supports a subset of hypotheses without supporting any strict subset of that set by lack of appropriate information [11]. A basic belief assignment (bba) is a function denoted m that assigns a value in $[0, 1]$ to every subset A of Θ .

This function m is defined here by:

$$m : 2^\Theta \rightarrow [0,1] \text{ such that}$$

$$m(\emptyset) = 0 \text{ and } \sum_{A \subseteq \Theta} m(A) = 1$$

The subsets A of the frame of discernment Θ which $m(A)$ are strictly positive, are called focal elements of the bba.

The credibility Bel and the plausibility Pl are defined by:

$$\text{Bel}(A) = \sum_{B \subseteq A} m(B)$$

$$\text{Pl}(A) = \sum_{A \cap B \neq \emptyset} m(B)$$

The quantity Bel(A) expresses the total belief fully committed to the subset A of Θ . Pl(A) represents the maximum amount of belief that might support the subset A.

Within the belief function model, it is easy to express the state of total ignorance. This is done by the so-called vacuous belief function which only focal element is the frame of discernment Θ . It is defined by [8]:

$$m(\Theta) = 1 \text{ and } m(A) = 0 \text{ for } A \neq \Theta.$$

Assessments of the bba are explained in [11, 13].

3.2 Combination

Let Bel₁ and Bel₂ be two belief functions induced by two distinct pieces of evidence. Let m₁ and m₂ denote their bba, respectively. The Dempster rule of combination aims at building the bba that represents the impact of the combined evidence. It is defined as [8]:

$$\begin{aligned} \forall A \subseteq \Theta, m(A) &= (m_1 \oplus m_2)(A) \\ &= K \cdot \sum_{B, C \subseteq \Theta: B \cap C = A} m_1(B) \cdot m_2(C) \end{aligned}$$

$$\text{where } K^{-1} = 1 - \sum_{B \cap C = \emptyset} m_1(B) \cdot m_2(C)$$

K is the normalization factor (we work here under the close world assumption [11]).

Dempster's rule of combination is a conjunctive rule: it builds the bba when both pieces of evidence are accepted. The dual of this conjunctive rule is the disjunctive rule of combination [10] 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 don't know which one. This rule is defined as following:

$$\forall A \subseteq \Theta, m_1 \vee m_2(A) = \sum_{B, C \subseteq \Theta: B \cup C = A} m_1(B) \cdot m_2(C)$$

These rules of combination are commutative and associative. Thus, the basic belief assignment resulting from the combination of several pieces of information can be computed easily by applying

repeatedly the rule and without worrying about the order of combination.

The conjunctive and disjunctive rules of combination generalize the AND and OR operators of set theory. To see it, consider m₁ and m₂ such that m₁(A) = 1 and m₂(B) = 1, what just translates the fact that the first piece of evidence states for sure that A holds, and the second that B holds. If we know that both pieces of evidence are to be accepted, then we know that A ∩ B holds (and indeed, m₁ ⊕ m₂(A ∩ B) = 1). If we know only that at least one piece of evidence is to be accepted and we do not know which one, then we only know that A ∪ B holds (and indeed, m₁ ∨ m₂(A ∪ B) = 1).

3.3 Decision making

The problem of decision making in the context of the TBM is solved in [11].

The TBM is based on a two level mental modals:

- 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.

The link between these two functions is achieved by the pignistic transformation that builds the pignistic probability function, denoted BetP, induced by a belief function. It is defined by:

$$\text{BetP}(B) = \sum_{A \subseteq \Theta} m(A) \frac{|B \cap A|}{|A|}, \text{ for all } B \subseteq \Theta.$$

4. Decision tree using the belief function theory

In this section, we detail our decision tree algorithm based on the belief function theory. First, we present the decision tree building phase, then the classification phase. The two phases will be illustrated by examples in order to understand their unfolding.

4.1 Decision tree building phase

In this part, we define the main parameters of a decision tree within the belief function framework, then we present our algorithm for building such decision trees.

Definition of decision tree parameters:

Our algorithm for building decision trees using the theory of belief functions is based on an extension of the ID3 algorithm [4] while taking into account the uncertainty of some parameters related to the classification problem. Thus, several differences will be noted in the definition of the hypotheses on these parameters and consequently in their treatment.

First, we have to define the structure of the training set under this uncertain framework. This set is generally composed of elements represented as pairs (attributes, class) where for each example, we know exactly the value of each one of its attributes and also its assigned class which is unique.

Unlike the standard training set, we assume that it may contain data where there is some uncertainty in the knowledge of the classes. In other words, each class of the training example may be uncertain or even unknown, whereas the values of the attributes characterizing each training example are known with certainty. Generalizing by accepting also uncertainty in the attributes' values is under development.

We propose to represent the uncertainty on the classes of any training example by a basic belief assignment defined on the set of classes related to the problem. This bba, generally given by an expert, represents the opinions-beliefs of this expert about the actual value of the class for each case in the training set.

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

- When we do not have any information about the classes of the example, the bba will be the vacuous belief function defined by:

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

- When the class of the example is perfectly known, it will be represented by the belief function:

$$m(C_i) = 1 \text{ and } m(C) = 0 \text{ for all } C \neq C_i, C \subseteq \Theta, \text{ where } C_i \text{ is a singular class.}$$

The latter case corresponds to the classical "certain" context.

Once the structure of the training set is described, the second important parameter to define in our algorithm is the attribute selection measure that will be used to find the appropriate test attribute at each decision node in the tree.

This measure enables us to quantify the power of discrimination of each attribute relatively to each class. It allows optimizing the tree. In the literature and even in practice, one of the selection measures most commonly applied in decision trees is the information gain criterion of Quinlan [4] [7].

Within our structure of the training set, we develop an attribute selection measure based on the gain criterion able to handle the uncertainty using the belief function framework.

Let Bel_j be the belief function defined on the set of possible classes. It represents the beliefs held by the expert about the actual value of the class to which an object I_j belongs.

Suppose a subset S of objects in the training set, and suppose we select randomly one object in S with equi-probability (generalization to more elaborated sampling schema is immediate). The belief function that represents our beliefs about the actual class to which this randomly selected individual belongs to is the average belief function taken over the object in S .

$$Bel_S(C) = \frac{\sum_{I_j \text{ in } S} Bel_j(C)}{|S|}$$

for any C subsets of $\Theta = \{C_1, \dots, C_n\}$.

Note that the bba and the pignistic probabilities related to this average belief function are respectively equal to the average of the basic belief assignments and the pignistic probabilities of the objects in S . For any C subsets of Θ ,

$$m_S(C) = \frac{\sum_{I_j \text{ in } S} m_j(C)}{|S|}$$

$$BetP_S(C) = \frac{\sum_{I_j \text{ in } S} BetP_j(C)}{|S|}$$

We propose the following steps to build the tree:

1. Compute the average pignistic probability function $BetP_T$ taken over the training set T . Then compute the entropy of the class distribution in T . This value $Info(T)$ is equal to:

$$Info(T) = - \sum_{i=1}^n BetP_T(C_i) \log_2 BetP_T(C_i)$$

Compute the information gain provided by each attribute A as:

$$\text{Gain}(T, A) = \text{Info}(T) - \text{Info}_A(T).$$

2. Our task is at first to define $\text{Info}_A(T)$ for each attribute. The idea is to apply the same procedure as in the computation of $\text{Info}(T)$, but restricting ourselves to the set of objects that share the same value for the attribute A and averaging these conditional information measures.

For each attribute value a_m , we build the subset T_m made of the cases in T whose value for the attribute is a_m . We compute the average belief function Bel_{T_m} , then apply the pignistic transformation to it in order to compute the pignistic probability BetP_{T_m} . From it, we compute $\text{Info}(T_m)$ where T_m represents the training subset when the value of the attribute A is equal to a_m .

3. $\text{Info}_A(T)$ will be equal to the weighed sum of the different $\text{Info}(T_m)$ relative to the considered attribute. These $\text{Info}(T_m)$ will be weighted by the proportion of each attribute value in the training set.

$$\begin{aligned} \text{Info}_A(T) &= \sum_{m=1}^k \frac{|T_m|}{|T|} \text{Info}(T_m) \\ &= - \sum_{m=1}^k \frac{|T_m|}{|T|} \sum_{i=1}^n \text{BetP}_{T_m}(C_i) \cdot \log_2 \text{BetP}_{T_m}(C_i) \end{aligned}$$

4. Once the different attribute information gains are computed, we choose the attribute with the highest value of the information gain.

In addition to the attribute selection measure, two other major parameters have to be defined:

- The partitioning strategy : Since we deal with symbolic attributes, it simply consists in creating an edge for each attribute value.

- The stopping criterion: It allows to stop the development of a path and to declare the node as a leaf. That is, it determines whether or not a training subset should be further divided. For our algorithm, we propose as a stopping criterion the following alternatives:

1. If the treated node includes only one object, then the node is declared as a leaf characterized by the same bba of this object which is already defined in the training set.

2. If there is no further attribute for testing or if the gain criterion for the remaining attributes is less than zero, then the node is a leaf where its bba will be the result of conjunctive combination of the objects' bba

belonging to the same leaf by applying Dempster's rule.

The choice of the conjunctive rule is based on the idea that every object in a leaf belongs to the same class. So in the simplified case where there are only two objects in a leaf and all we know is that one is A or B and the other B or C, then we can conclude that the objects of that leaf are B's, hence the conjunctive rule.

Unlike the standard decision tree where each leaf is labeled by a unique class, our method assigns to each leaf a bba expressing the mass of beliefs on the different classes of the frame of discernment.

Decision tree building algorithm:

Let T be a training set composed by objects characterized by l symbolic attributes (A_1, A_2, \dots, A_l) and that may belong to the set of classes $\Theta = \{C_1, C_2, \dots, C_n\}$. For each object I_j ($j = 1.. p$) of the training set will correspond a basic belief assignment 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 [4]. Their steps are described as follows:

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

2. Verify if this node satisfies or not the stopping criterion:

- If yes, declare it as a leaf node and compute its corresponding bba as we mentioned in the last section.

- If not, look for the attribute having the highest information gain. This attribute will be designed as the root of the decision tree related to the whole training set.

3. Apply the partitioning strategy by developing an edge for each attribute value chosen as a root. This partition leads to several training subsets.

4. Repeat the same process for each training subset from the step 2 while verifying the stopping criterion. If this latter is satisfied, declare the node as a leaf and compute its assigned bba, else repeat the same process.

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

We have to mention that we get the same results as ID3 if all the bba are 'certain'. That is when the class assigned for each training example is unique and known with certainty.

Example 1:

Now, we present a simple example illustrating our decision tree building algorithm within a belief function framework.

Let T be a small training set (see table 1). It is composed of five objects characterized by three symbolic attributes defined as following:

Eyes = {Brown, Blue}; Hair = {Dark, Blond};

Height = {Short, Tall}

As we work in a supervised learning context, the possible classes are already known. We denote them by C_1, C_2 and C_3 .

For each object I_j ($j = 1..5$) belonging to the training set T, we assign a bba m_j expressing our beliefs on its actual class. These functions are defined on the same frame of discernment $\Theta = \{C_1, C_2, C_3\}$.

Table1: Training set T

I_j	Eyes	Hair	Height	Classes
I_1	Brown	Dark	Short	m_1
I_2	Brown	Dark	Tall	m_2
I_3	Blue	Blond	Tall	m_3
I_4	Blue	Dark	Tall	m_4
I_5	Blue	Dark	Short	m_5

where

$$m_1(C_1) = 0.3; m_1(C_1 \cup C_2) = 0.4; m_1(\Theta) = 0.3;$$

$$m_2(C_2) = 0.5; m_2(C_1 \cup C_2) = 0.2; m_2(\Theta) = 0.3;$$

$$m_3(C_1) = 0.8; m_3(\Theta) = 0.2;$$

$$m_4(C_2) = 0.1; m_4(C_3) = 0.3; m_4(C_2 \cup C_3) = 0.2; m_4(\Theta) = 0.4;$$

$$m_5(C_2) = 0.7; m_5(\Theta) = 0.3;$$

In order to find the root relative to the decision tree, we have to compute the average belief function Bel_T related to the whole training set T. Bel_T and its corresponding bba m_T are presented in the following table (see table 2):

Table 2: Computation of Bel_T and m_T

	\emptyset	C_1	C_2	C_3	$C_1 \cup C_2$	$C_1 \cup C_3$	$C_2 \cup C_3$	Θ
Bel_T	0	0.22	0.26	0.06	0.6	0.28	0.36	1
m_T	0	0.22	0.26	0.06	0.12	0	0.04	0.3

The pignistic transformation of m_T gives as results: $BetP_T(C_1) = 0.38; BetP_T(C_2) = 0.44; BetP_T(C_3) = 0.18;$

Hence

$$Info(T) = - \sum_{i=1}^3 BetP_T(C_i) \log_2 BetP_T(C_i) = 1.496$$

Once the entropy related to the whole set T is calculated, the second step is to find the information gain of each attribute in order to choose the root of the decision tree.

Let's illustrate the computation for the eye attribute.

Let $Bel_{T_{br}}$ be the average belief function relative to the objects belonging to T and having brown eyes whereas, $Bel_{T_{bl}}$ for the ones having blue eyes. $m_{T_{br}}, m_{T_{bl}}, BetP_{T_{br}}$ and $BetP_{T_{bl}}$ are respectively the bba and the pignistic probability relative to the values brown and blue of the eyes (see table 3 and table 4).

Table 3: Computation of $Bel_{T_{br}}, m_{T_{br}}, Bel_{T_{bl}}, m_{T_{bl}}$

	\emptyset	C_1	C_2	C_3	$C_1 \cup C_2$	$C_1 \cup C_3$	$C_2 \cup C_3$	Θ
$Bel_{T_{br}}$	0	0.15	0.25	0	0.7	0.15	0.25	1
$m_{T_{br}}$	0	0.15	0.25	0	0.3	0	0	0.3
$Bel_{T_{bl}}$	0	0.27	0.27	0.1	0.54	0.37	0.43	1
$m_{T_{bl}}$	0	0.27	0.27	0.1	0	0	0.06	0.3

Table 4: Computation of $BetP_{T_{br}}, BetP_{T_{bl}}$

	C_1	C_2	C_3
$BetP_{T_{br}}$	0.4	0.5	0.1
$BetP_{T_{bl}}$	0.37	0.4	0.23

$$Info_{eyes}(T) = - \frac{2}{5} \sum_{i=1}^3 BetP_{T_{br}}(C_i) \cdot \log_2 BetP_{T_{br}}(C_i) -$$

$$\frac{3}{5} \sum_{i=1}^3 BetP_{T_{bl}}(C_i) \cdot \log_2 BetP_{T_{bl}}(C_i) = 1.4732$$

$$\text{Thus Gain}(T, \text{Eyes}) = Info(T) - Info_{eyes}(T) = 0.0228;$$

By similar analysis for the hair and height attributes, we get:

$$\text{Gain}(T, \text{Hair}) = 0.1876; \text{Gain}(T, \text{Height}) = 0.0316;$$

According to the gain criterion, the hair attribute will be chosen as the root of the decision tree and branches are created below the root for each of its possible value (Dark, Blond).

So, we get the following decision tree (see figure 1):

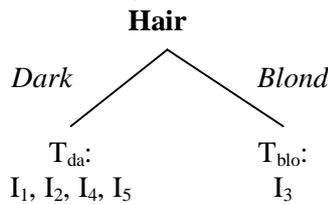


Figure 1: First generated decision tree

We notice that the training subset T_{blo} contains only one example, thus the stopping criterion is satisfied. The node relative to T_{blo} is therefore declared as a leaf defined by the bba m_3 of the example I_3 .

For the subset T_{da} , we apply the same process as we did for T until the stopping criterion holds.

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

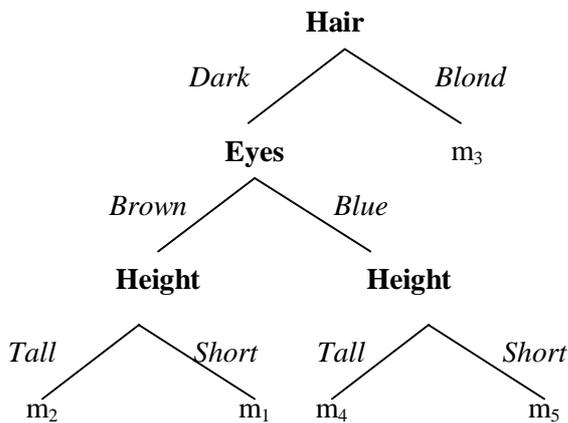


Figure 2: The final decision tree

4.2 Case's classification

Once the decision tree is constructed, the following phase will be the classification of unseen examples referring to as new objects.

On one hand, our algorithm is able to ensure the standard classification where the unseen example attribute values are assumed to be certain.

As in an ordinary tree, it consists on 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 decision tree, the unseen example classes will be defined by a basic belief assignment related to the reached leaf.

In order to make a decision and to get the probability of each singular class, we propose to apply the pignistic transformation to the basic belief assignment related to the reached leaf, and to use this probability distribution to compute the expected utilities required for optimal decision making.

On the other hand, as we deal with an uncertain context, our classification method allows also classifying unseen examples characterized by uncertainty in the values of their attributes.

In our method, we assume that new examples to classify are not only described by certain attribute values but may also be characterized by means of disjunction values for some attributes. They may even have attributes with unknown values.

The idea to classify such objects is to look for the leaves that the given example may belong by tracing out all the possible paths induced by the different attribute values. In the case of unknown values, all the branches relative to the considered attribute will be taken into account.

As a consequence, the unseen new example may belong to many leaves where each one is characterized by a basic belief assignment function. These bba must be combined in order to get beliefs on the example's classes. The disjunctive rule of combination developed by Smets [10] seems offering a suitable context since it supposes that at least one path is true.

Indeed consider the simplified case where there are two leaves in which the new object can belong, and all objects in leaf 1 are A's, and all those in leaf 2 are B's. Then all we can conclude is that the new object is A or B, hence the disjunctive rule.

The induced bba from the disjunctive rule can be transformed into a probability function by applying the pignistic transformation. This allows knowing the probability that the new example belongs to each singular class of the given problem.

Example 2:

Let's continue example 1 and assume that an unseen example is characterized by:

Hair = Dark; Eyes = Blue or Brown; Height = Tall.

Using the decision tree (see figure 2) relative to the training set T , gives us two possible leaves for this case:

- The first leaf characterized by m_2 as a b.p.a. This leaf is induced by the path corresponding to dark hair, brown eyes and tall as height.

- The second is the one corresponding to the path defined by dark hair, blue eyes and tall as height. This leaf is labeled by the b.p.a m_4

By applying the disjunctive rule of combination, we get $m_{24} = m_2 \vee m_4$ defined by:

$$m_{24}(C_2) = 0.05; m_{24}(C_1 \cup C_2) = 0.02; m_{24}(C_2 \cup C_3) = 0.25; m_{24}(\Theta) = 0.68;$$

Thus, the unseen example classes are described by m_{24} . Applying the pignistic transformation on m_{24} gives us:

$$\text{BetP}_{24}(C_1) = 0.24; \text{BetP}_{24}(C_2) = 0.41; \text{BetP}_{24}(C_3) = 0.35;$$

It seems that the most probable class for this example to belong is C_2 with the probability of 0.41.

5 Conclusion

In this paper, we propose an algorithm to generate a decision tree under uncertainty within the belief function framework.

The interest of the TBM appears essentially in its ability to cope with partial ignorance, and at the level of the leaves conjunctive and disjunctive rules can be used in a coherent way as they provide conjunctive and disjunctive aggregation rules.

First, we have interested to the decision tree building phase by taking into consideration the uncertainty characterized the classes of the training examples. Next, we have ensured the classification task of new examples where some of their attribute values are assumed to be uncertain.

Either in the decision tree building task or in the classification task, the uncertainty is handled within the theory of belief functions which presents a convenient framework for coping with lack of information.

Our future researches aims at extending this method in order to treat more uncertainty especially those held in the attribute values.

References

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

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

[3] R. Lopez De Mantaras "A distance-based attribute selection measure for decision tree induction" Machine learning 6, pp 81-92, 1991.

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

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

[6] J. R. Quinlan "Decision trees and decision making" IEEE Transactions on Systems, Man and Cybernetics, Vol 20 N°2, pp 339-346 March/April, 1990.

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

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

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

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

[11] P. Smets, R. Kennes "The transferable Belief Model "Artificial Intelligence Vol 66, pp 191-234, 1994 .

[12] 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, Kluwer, Dordrecht, 1998, pp 267-301.

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

[14] P. E. Utgoff "Incremental induction of decision trees" Machine Learning, 4, pp 161-186, 1989.