Decision tree classifiers for evidential attribute values and class labels

Asma Trabelsi, Zied Elouedi, Eric Lefevre

Decision tree classifiers for evidential attribute values and class labels

Asma Trabelsi[a,b], Zied Elouedi[a], Eric Lefevre[b]

[a] *Université de Tunis, Institut Supérieur de Gestion de Tunis, LARODEC, Tunis, Tunisia*
[b] *Université d'Artois, EA 3926, Laboratoire de Génie Informatique et d'Automatique de l'Artois (LGI2A), 62400 Béthune, France*

Abstract

Decision trees are well-known machine learning techniques for solving complex classification problems. Despite their great success, the standard decision tree algorithms do not have the ability to process imperfect knowledge, meaning uncertain, imprecise and incomplete data. In this paper, we develop new decision tree approaches to cope with data that have uncertain attribute values and class labels. More concretely, we tackle the case where the uncertainty is represented and managed through the evidence theory.

Keywords: Decision tree classifier, imperfect data, evidence theory.

1. Introduction

Decision trees are commonly seen efficient machine learning techniques that are widely used in several fields, notably in artificial intelligence [1]. Their success is adequately explained by their ability to provide simple representations that are easily understandable by experts and even by ordinary users

Email addresses: trabelsyasma@gmail.com (Asma Trabelsi), zied.elouedi@gmx.fr (Zied Elouedi), eric.lefevre@univ-artois.fr (Eric Lefevre)

(i.e., non-expert users).

In several real world domains (e.g. medicine, fault detection and object recognition), data may suffer from imperfection due to some factors such as randomness and data incompleteness. Data imperfection may take different forms: it can be part either of the attribute values, the class labels or both of them. Since the standard decision tree versions cannot handle such kind of data, probability decision trees have been suggested [2]. Although the probability theory is widely used for modeling uncertainty, several researchers have proven that probability cannot always be the adequate tool for representing uncertain and incomplete data [3]. This shortcoming has led to the introduction of fuzzy decision trees [4], the possibilistic decision trees [5], the uncertain decision trees [6, 7, 8]. Other theories have been proposed to deal with uncertain knowledge, notably the belief function theort also called the evidence theory. It has the advantage to represent all kinds of knowledge availability [9], the process of incorporating belief function theory within decision tree techniques has been extensively studied [10, 11, 12, 13, 14, 15, 16, 17]. Despite its benefits, decision trees that handle data with uncertain attribute values and class labels have not attracted attention from the community.

In this paper, we propose new decision tree classifier approaches to cope with data imperfection pervading both attribute values and class labels. The remaining of this paper is organized as follows. Section 2 is devoted to highlighting the fundamental concepts of the belief function theory. We recall, in Section 3, some basic concepts of standard decision tree classifiers. Section 4 is dedicated to describe machine learning algorithms within evidential data.

Section 5 details our decision trees procedure. In Section 6, we detail the parameters enabling the construction of our proposed decision tree classifier techniques. Our experimentations are described in Section 7. We draw our conclusions and discuss future directions in Section 8.

2. Evidence theory

The evidence theory is seen as a very effective and efficient framework to represent and manage uncertain knowledge. In this section, we provide a brief overview of the fundamental concepts of this theory as introduced by Smets [18] with his Transferable Belief Model (TBM).

2.1. Knowledge representation

Let $\Theta = \{\theta_1, \theta_2, \ldots, \theta_N\}$ denotes the frame of discernment including a finite non empty set of N elementary hypotheses. An expert's belief over the subsets of the frame of discernment Θ is represented by the so-called basic belief assignment (bba) denoted by m. It is carried out in the following manner:

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

The basic belief mass (bbm), denoted by $m(A)$, implies the degree of belief exactly assigned to the event A. Each subset A of 2^Θ having fulfilled:

$$m(A) > 0 \tag{2}$$

is called a focal element.

From a mass function m, we can define the so called belief function denoted by *bel*. It reflects the sum of beliefs exactly committed to every subset of A by m [19]. It is set to:

$$bel : 2^{\Theta} \rightarrow [0,1]$$
$$bel(A) = \sum_{\emptyset \neq B \subseteq A} m(B). \tag{3}$$

2.2. Combining information sources

Let m_1 and m_2 be two bbas induced from two independent information sources and defined in the same frame of discernment Θ. Several combination rules have been proposed to combine several bbas. One of the most widely used is the conjunctive rule [20]. This combination rule combines two bbas provided by reliable and distinct information sources. The resulting bba, denoted by $m_1 \textcircled{\cap} m_2$, is defined by:

$$(m_1 \textcircled{\cap} m_2)(A) = \sum_{B,C \subseteq \Theta : B \cap C = A} m_1(B).m_2(C). \tag{4}$$

Several real world applications require to combine bbas defined on different frames of discernment. Assuming that Θ_1 and Θ_2 are two frames of discernment, the idea consists of extending Θ_1 and Θ_2 to a joint frame of discernment $\Theta = \Theta_1 \times \Theta_2$. The extended mass function of m_1 which is defined on Θ_1 and whose focal elements are the cylinder sets of the focal elements of m_1 is computed as follows:

$$\begin{aligned} m^{\Theta_1 \uparrow \Theta}(X) &= m_1(Y) \qquad where\ X = Y \times \Theta_2,\ Y \subseteq \Theta_1 \\ m^{\Theta_1 \uparrow \Theta}(X) &= 0\ otherwise. \end{aligned} \tag{5}$$

2.3. Discounting

Information sources, within the belief function theory framework, may be quantified by reliability rates in the range between 0 and 1. Indeed, when a bba is induced from not fully reliable information sources, a discounting process is necessary to update beliefs. Let m be a bba induced from an information source with a reliability rate $1-\alpha$. The discounted bba m^{α} is obtained as follows [21]:

$$\begin{aligned} m^{\alpha}(A) =& (1-\alpha)m(A) \text{ for } A \subset \Theta \\ m^{\alpha}(\Theta) =& \alpha + (1-\alpha)m(\Theta). \end{aligned} \tag{6}$$

2.4. Decision making

Decision making aims to select the most reasonable hypothesis for a given problem. Several functions have been introduced for decision making within the belief function framework. In the Transferable Belief Model (TBM), the pignistic probability is commonly used to make a decision from a bba [22]:

$$BetP(A) = \sum_{B \subseteq \Theta} \frac{|A \cap B|}{|B|} \frac{m(B)}{1 - m(\emptyset)} \ \forall \ A \in \Theta \tag{7}$$

where

$$\sum_{A \in \Theta} BetP(A) = 1. \tag{8}$$

2.5. The dissimilarity between two bbas defined on the same space of discernment

Several measures have been proposed to compute the dissimilarity between two given bbas [23, 24]. One of the earliest and best-known measures is the

Jousselme distance. Formally, the Jousselme distance, for two given bbas m_1 and m_2, is defined as [23]:

$$d(m_1, m_2) = \sqrt{\frac{1}{2}(\overrightarrow{m_1} - \overrightarrow{m_2})^T . D . (\overrightarrow{m_1} - \overrightarrow{m_2})} \tag{9}$$

where

$$\overrightarrow{m_1} = \begin{pmatrix} m_1(\emptyset) \\ \vdots \\ m_1(\Theta) \end{pmatrix} \qquad and \qquad \overrightarrow{m_2} = \begin{pmatrix} m_2(\emptyset) \\ \vdots \\ m_2(\Theta) \end{pmatrix}$$

The Jaccard similarity measure D is set to:

$$D(A, B) = \begin{cases} 1 & \text{if } A{=}B{=}\emptyset \\ \dfrac{|A \cap B|}{|A \cup B|} & \forall\ A{,}B \in 2^{\Theta}. \end{cases} \tag{10}$$

Note that $d(m_1, m_2) \in [0, 1]$. A value of 1 reflects that the two bbas m_1 and m_2 are in total disagreement, while a value of 0 means that $m_1 = m_2$.

3. Decision tree classifier

Decision trees are recognized among the most effective and efficient machine learning approaches and they have been successfully applied to solve real world problems within the artificial intelligence field. This success is mainly due to their great ability for solving complex problems through human-readable and computer-readable graphical representations. A plethora of algorithms have been introduced to construct decision trees from a given

training set and to ensure the classification of query instances [25, 26, 27]. The most used algorithms follow a Top Down Induction of Decision Tree approach (TDIDT) that consists on a recursive divide and conquer strategy by following the steps below:

- select, through the use of an attribute selection measure, the attribute that enables the best possible partitioning of the training set;
- split the current training data into training subsets according to the selected attribute values.
- nominate a training subset as a leaf when a stopping criterion is reached.

As regards the attribute selection process, several measures have been proposed in the literature [28, 26, 27]. The information gain, measuring the efficiency of an attribute when classifying the training instances, is one among the best known and most widely used measures. Given a training data S and an attribute A, the information gain will be set to:

$$Gain(S, A) = Info(S) - Info_A(S) \tag{11}$$

where

$$Info(S) = -\sum_{i=1}^{Q} p_i.log_2 p_i \tag{12}$$

and

$$Info_A(S) = -\sum_{v \in Domain(A)} \frac{|S_v^A|}{|S|} \tag{13}$$

where p_i reflects the proportion of objects having θ_i as class (i.e. $i \in \{1, \ldots, Q\}$) and S_v^A corresponds to the training subsets for which the attribute A has v as value.

One major limitation of this measure is that the attributes with the largest values are the most promoted ones [27]. This had led to the introduction of the *GainRatio* measure used in the C4.5 algorithm [26, 27]. It is given as follows:

$$GainRatio(S, A) = \frac{Gain(S, A)}{SplitInfo(A)} \tag{14}$$

where

$$SpliInfo(A) = \sum_{v \in Domain(A)} \frac{|S_v^A|}{|S|}.log_2 \frac{|S_v^A|}{|S|}. \tag{15}$$

4. Classification from evidential data

Evidential databases allow to process imperfect knowledge by expressing imperfect knowledge through the belief function theory. Each object O_j within an evidential database is described by n evidential attributes $A = \{A_1, \ldots, A_n\}$. Each attribute A_k (i.e. $k \in \{1, \ldots, n\}$) has a domain of discrete values denoted by Θ^{A_k}, and an evidential class label having $\Theta = \{\theta_1, \ldots, \theta_Q\}$ as domain (i.e. Q represents the total number of classes). The idea behind evidential databases consists of representing all kinds of data availability, including total certainty as well as totally and partially ignorance:

- **Total certainty**: A certain bba, is a bba that has a singleton as its

unique focal element, and it is used to represent the state of total certainty.

- **Total ignorance**: In case of total ignorance, a vacuous bba refers to a bba having Θ as its unique focal element.

- **Partially ignorance**: Regarding the case of partially ignorance, a quantity of belief has to be assigned to a subset of the frame of discernment Θ.

An example of evidential databases is given below.

Example 1. Assuming a credit risk management problem. Overall, a bank loan officer has to predict the customer profitability levels Θ={Good, Moderate, Bad} on the basis of some parameters (Attributes). To put it simply, in this example, Table 1 describes the data knowledge for training, where we relied on three characteristics:

- Income with possible values Θ^{Income}={No, Low, Average, High}.
- Property: This attribute reflects whether the loan requested by the client is greater or less than its property value and consequently it takes values as $\Theta^{Property}$={Greater, Less}.
- Unpaid Credit: This attribute providing information about client's unpaid Credit with two possible values $\Theta^{UnpaidCredit}$={Yes, No}.

Table 1: Uncertain training data within the belief function framework

width=1.2center

O	Income	Property	Unpaid Credit
O_1	$m_1^{Income}(\{High\}) = 1$	$m_1^{Property}(\{Greater\}) = 0.6$ $m_1^{Property}(\{Less\}) = 0.3$ $m_1^{Property}(\Theta^{Property}) = 0.1$	$m_1^{UnpaidCredit}(\{Yes\}) =$
O_2	$m_2^{Income}(\{Average\}) = 1$	$m_2^{Property}(\{Greater\}) = 1$	$m_2^{UnpaidCredit}(\{No\}) = 1$
O_3	$m_3^{Income}(\{Low\}) = 1$	$m_3^{Property}(\{Less\}) = 0.5$ $m_3^{Property}(\Theta^{Property}) = 0.5$	$m_3^{UnpaidCredit}(\{Yes\}) =$
O_4	$m_4^{Income}(\{No\}) = 1$	$m_4^{Property}(\{Less\}) = 1$	$m_4^{UnpaidCredit}(\{No\}) = 1$
O_5	$m_5^{Income}(\{Low\}) = 1$	$m_5^{Property}(\{Greater\}) = 0.8$ $m_5^{Property}(\Theta^{Property}) = 0.2$	$m_5^{UnpaidCredit}(\{No\}) = 1$
O_6	$m_6^{Income}(\{High\}) = 1$	$m_6^{Property}(\{Greater\}) = 0.2$ $m_6^{Property}(\{Less\}) = 0.7$ $m_6^{Property}(\Theta^{Property}) = 0.1$	$m_6^{UnpaidCredit}(\{Yes\}) =$

We intend, in this paper, to construct decision tree classifiers from evidential data using the following notations:

- $T = \{O_1, \ldots O_M\}$: a given training set composed by M objects O_j; $j = \{1,\ldots,M\}$.
- S: a subset of objects belonging to the training set T.
- A=$\{A_1,\ldots,A_n\}$: the set of n attributes.
- Θ^{A_k}: set of all possible values v of an attribute $A_k \in A$ where $k =$

$\{1, \ldots, n\}$.

- $\Theta = \{\theta_1, \ldots, \theta_Q\}$: represents the Q possible classes of the classification problem.
- $S_v^{A_k}$: subset of objects from S having $v \in \Theta^{A_k}$ as value.
- $m_j^{\Theta^{A_k}}(v)$: denotes the bbm assigned to the hypothesis that the actual attribute value of the object O_j belongs to $v \subseteq \Theta^{A_k}$.
- $m^{\Theta^{A_v^k}}$: is the certain bba corresponding to the attribute A_k and having v as its unique focal element.
- m_j^{Θ}: corresponds to the bba relative to the class of the object O_j.
- $L = \{L_1, \ldots, L_F\}$: represents the F generated leaves when building the decision tree.

5. Decision tree procedure

This section presents the two main levels enabling the use of our belief decision tree classifiers: the construction and the classification levels. We provide, in the following, a description of each level.

5.1. Construction level

The construction of our belief decision trees follows the same Quinlan algorithm steps [27]. In fact, it requires a top down approach as the standard case. Assume that T is our training set, the different steps of our decision tree learning algorithm are given as follows:

1. We start by creating the root node from the whole training set T.
2. We check if the root node satisfies any stopping criteria.
   - If one stopping criterion is reached, the treated node will be declared as a leaf for which we compute the probability distribution over the set of classes.
   - If not, we pick out the attribute that maximizes a chosen attribute selection measure. The chosen one will be the root node of our decision tree relative to the set T.
3. We create a branch for each value v of the attribute A_k chosen as a root. This partitioning step leads to several subsets $S_v^{A_k}$ where each one contains as much as possible homogenous objects according to the attribute value v.
4. We restart the same process from level 2 until all nodes are considered as leaves.

5.2. Classification level

Concerning the classification step, we propose to classify objects described by evidential attributes modeled with bbas. Let M' be the total number of testing instances O_j ($j = \{1, \ldots, M'\}$) and $A = \{A_1, \ldots, A_n\}$ be the set of n attributes characterizing our testing instances. The global frame of discernment relative to all the attributes, denoted by Θ^A, is equal to the cross product of the different Θ^{A_k}:

$$\Theta^A = \underset{k=1,\ldots,n}{\times} \Theta^{A_k}. \tag{16}$$

Since objects are described by a combination of values where each of them corresponds to one attribute, we first have to compute for each object to be classified the joint bba expressing beliefs on its attribute values. To reach our ultimate goal, we proceed as follows:

- Firstly, we extend the different bbas $m_j^{\Theta^{A_k}}$ to the global frame of attributes Θ^A (see Equation 5). Thus, we get the different bbas $m_j^{\Theta^{A_k}\uparrow\Theta^A}$.

- Then, we combine the different extended bbas using the conjunctive combination rule as follows:

$$m_j^{\Theta^A} = \bigcirc\!\!\!\!\!\!\bigcap_{k=1,\ldots,n} m_j^{\Theta^{A_k}\uparrow\Theta^A}. \tag{17}$$

If our joint bba $m_j^{\Theta^A}$ is computed, we move on to evaluate the belief function $bel_j^{\Theta}[x]$ of each focal element x relative to the object O_j. It will be noted that the computation of this function depends mainly on the focal elements of the bba m^{Θ^A} and on the subset x. This dependency is expressed in the following:

- When x is a singleton, the belief function $bel_j^{\Theta}[x]$ will be equal to the belief function corresponding to the leaf to which the focal element is attached;

- If not, we explore all possible paths corresponding to this combination of values. There are two possible cases:

  - Case 1: All paths lead to the same leaf. In this case, the $bel_j^{\Theta}[x]$ will be equal to the leaf's belief function.

– Case 2: Paths lead to distinct leaves. In this case, $bel_j^{\Theta}[x]$ will correspond to the disjunctive combination [29] of each leaf's belief function through the disjunctive rule.

- The belief function of every query test O_j has to be computed by averaging each focal element x using m^{Θ^A} when relied on the generalized Bayesian theorem [30]:

$$bel_j^{\Theta}[m^{\Theta^A}](\theta) \quad = \quad \sum_{x \subseteq \Theta^A} m^{\Theta^A}(x).bel_j^{\Theta}[x](\theta) \quad \text{for } \theta \subseteq \Theta \quad (18)$$

then

$$m_j^{\Theta}[m^{\Theta^A}](\theta) \quad = \quad \sum_{B \subseteq \theta} (-1)^{|\theta|-|B|}.bel_j^{\Theta}[m^{\Theta^A}](B) \quad \forall\, \theta \subseteq \Theta,\, \theta \neq \emptyset \quad (19)$$

Note that $m_j^{\Theta}[m^{\Theta^A}](\theta)$ reflects the degree of belief allocated to the class θ of the query test O_j. In order to make the decision, the mass function will be transformed into probability measure through the pignistic probability. The class with the highest pignistic probability will be considered as the predicted class of the object O_j.

6. Novel belief decision tree classifiers

In this paper, our aim consists of designing two decision tree classifiers for addressing evidential data, namely *GainRatio* Belief Decision Tree (*GR*-BDT) and $DiffRatio$ Belief Decision Tree (*DR*-BDT).

6.1. GainRatio Belief Decision Tree (GR-BDT)

This subsection is devoted to highlighting the main parameters enabling the construction of the *GR*-BDT classifier which mainly includes the attribute selection measure, the partitioning strategy, the stopping criteria and the structure of leaves.

6.1.1. Attribute selection measure

The attribute selection measure is considered as one of the major parameters ensuring decision tree construction. It consists of choosing, for each decision node of the tree, the attribute test that will better separate the training instances into homogenous subsets. For the Gain Ratio approach, we have relied on the entropy measure that is calculated from the average probability obtained from the set of objects in the node. To choose the most appropriate attribute, we propose the following steps:

1. We compute the pignistic probability relative to each instance O_j belonging to the training set:

$$BetP^{\Theta}[O_j](\theta_i) = \sum_{\theta_i \subseteq B; B \subseteq 2^{\Theta}} \frac{1}{|B|} . \frac{m_j^{\Theta}(B)}{1 - m_j^{\Theta}(\emptyset)}. \tag{20}$$

2. We compute the average probability relative to each class by taking into consideration the objects in the set S (the learning set for which we look to identify the best attribute to split on). This function is set to:

$$BetP^{\Theta}[S](\theta_i) = \frac{1}{\sum_{O_j \in S} P_j^S} \sum_{O_j \in S} P_j^S . BetP^{\Theta}[O_j](\theta_i) \tag{21}$$

where $BetP^{\Theta}[O_j](\theta_i)$ reflects the membership probability of the object O_i to the class θ_i and P_j^S corresponds to the probability that the object

O_j belongs to the subset S. Assuming that the attributes are independent and assuming that $S \neq T$, the probability P_j^S will be equal to the product of the different pignistic probabilities induced from the attribute bba's corresponding to the object O_j and enabling this latter to belong to the node S. Let $A_B = \{A_1, \ldots, A_O\} \in A$ with values $V_B = \{v_1, \ldots, v_O\}$ be the set of attributes leading to the branch S, the probability P_j^S will be set to:

$$P_j^S = \prod_{A_o \in A_B} BetP^{\Theta^{A_o}}[O_j](v_o) \tag{22}$$

3. Inspired by the Shannon measure of uncertainty [31], we compute the entropy $Info(S)$ of the average probabilities in S which is set to:

$$Info(S) = -\sum_{i=1}^{Q} BetP^{\Theta}[S](\theta_i).log_2 BetP^{\Theta}[S](\theta_i). \tag{23}$$

4. Assuming an attribute A_k, for each value $v \in \Theta^{A_k}$, we define the subset $S_v^{A_k}$ which is composed with objects having v as their value. Since the attribute A_k's values may be uncertain, the subset $S_v^{A_k}$ will contain objects O_j such that $BetP^{\Theta^{A_k}}[O_j](v) \neq 0$.

5. We compute, for objects in the subset $S_v^{A_k}$, the weighted average pignistic probability $BetP^{\Theta}[S_v^{A_k}]$ where $v \in \Theta^{A_k}$ and $A_k \in A$. It will be set to:

$$BetP^{\Theta}[S_v^{A_k}](\theta_i) = \frac{1}{\sum_{O_j \in S_v^{A_k}} P_j^{S_v^{A_k}}} \sum_{O_j \in S_v^{A_k}} P_j^{S_v^{A_k}}.BetP^{\Theta}[O_j](\theta_i) \tag{24}$$

where $P_j^{S_v^{A_k}}$ is the probability of the object O_j belongs to the subset $S_v^{A_k}$ having v as a value of the attribute A_k (its computation is done in the same manner as the computation of P_j^S).

6. We compute $Info_{A_k}(S)$ as discussed by Quinlan [26], while using the pignistic probability instead of the proportions. We get:

$$Info_{A_k}(S) = \sum_{v \in \Theta^{A_k}} \frac{\sum_{O_j \in S_v^{A_k}} P_j^{S_v^{A_k}}}{\sum_{O_j \in S} P_j^S} . Info(S_v^{A_k}) \tag{25}$$

where $Info(S_v^{A_k})$ is computed from Equation 23.

7. We compute the *GainRatio* relative to the attribute A_k:

$$GainRatio(S, A_k) = \frac{Info(S) - Info_{A_k}(S)}{SplitInfo(S, A_k)} \tag{26}$$

where the $SplitInfo$ value is defined as follows:

$$SplitInfo(S, A_k) = - \sum_{v \in \Theta^{A_k}} \frac{\sum_{O_j \in S_v^{A_k}} P_j^{S_v^{A_k}}}{\sum_{O_j \in S} P_j^S} . log_2 \frac{\sum_{O_j \in S_v^{A_k}} P_j^{S_v^{A_k}}}{\sum_{O_j \in S} P_j^S}. \tag{27}$$

8. We repeat the same process for each attribute $A_k \in A$ (from step 3 to step 7) and then we select the one that has the maximum *GainRatio*.

Example 2. Let us consider the training data given in Example 1. By relying on the *GainRatio* measure, we try to illustrate the first attribute selection process when S=T. Note that, in the case of S=T, P_j^S equals 1 for all $j \in \{1, \ldots, 6\}$. The probability of belonging of an object O_j to a subset $S_k^{A_k}$ are given in Table 2.

- We start by computing the entropy $Info(T)$, using Equation 23. It is computed using the average pignistic probability that reflect the membership of objects to classes *Good*, *Moderate* or *Bad* (see Table 3). The entropy will then be set to:

$$\begin{aligned} Info(T) = & - 0.41.log_2 0.41 - 0.22.log_2 0.22 - 0.37.log_2 0.37 \\ = & 1.5426 \end{aligned}$$

Table 2: The probability of belonging of objects in terms of the attribute values.

	Outlook			
	$P^{S^{Income}_{High}}$	$P^{S^{Income}_{Average}}$	$P^{S^{Income}_{Low}}$	$P^{S^{Income}_{No}}$
O_1	1	0	0	0
O_2	0	1	0	0
O_3	0	0	1	0
O_4	0	0	0	1
O_5	0	0	1	0
O_6	1	0	0	0

	Temperature	
	$P^{S^{Property}_{Greater}}$	$P^{S^{Property}_{Less}}$
O_1	0.65	0.35
O_2	1	0
O_3	0.25	0.75
O_4	0	1
O_5	0.9	0.1
O_6	0.25	0.75

	Humidity	
	$P^{S^{UnpaidCredit}_{Yes}}$	$P^{S^{UnpaidCredit}_{No}}$
O_1	1	0
O_2	0	1
O_3	1	0
O_4	0	1
O_5	0	1
O_6	1	0

Table 3: Pignistic probability computed from bba of *Class* column of Table 1

	Good	Moderate	Bad
$BetP^{\Theta}(T)$	0.41	0.22	0.37

- Subsequently, we move on to compute $Info_{Income}(T)$, $Info_{Property}(T)$ and $Info_{UnpaidCredit}(T)$ through Equation 25. We get:
  - $Info_{Income}(T) = 0.9738$
  - $Info_{Property}(T) = 1.3959$
  - $Info_{UnpaidCredit}(T) = 1.4644$
- Then, we compute the Information Gain for each attribute:
  - $GainRatio(Income) = 0.2965$
  - $GainRatio(Property) = 0.1467$
  - $GainRatio(UnpaidCredit) = 0.0782$

  According to the yielded results, we can deduce that the attribute $Income$ is the one that maximizes the $GainRatio$. So, it will be chosen as the root node of the tree.

6.1.2. Partitioning strategy

The splitting strategy consists of dividing the training set according to the values of the chosen attribute A_k, meaning that a branch will be associated to each value v of the chosen attribute and each edge will contain a subset

$S_v^{A_k}$ from S. Since we handle data with evidential attributes, each training object may belong to more than one subset. Indeed, each training object may belong to more than one branch with a membership probability computed in terms of the pignistic probability. To put it simply, a given object O_j has to be assigned to each branch having v as value and should satisfy $BetP^{\Theta^{A_k}}[O_j](v) \neq 0$.

6.1.3. Stopping criteria

The stopping criteria are similar to those used by the standard decision tree. There exist mainly four stopping strategies:

1. Only one instance is part to the treated node.
2. Instances of the treated node belong to the same class.
3. There is no further attribute for checking.
4. The remaining attributes have *GainRatio* equal or less than zero.

6.1.4. Structure of leaves

It is important to underline that leaves for our constructed decision trees include objects with different class labels. As a matter of fact, each leaf will be represented by a belief function computed from the probability of objects belonging to that leaf:

$$m^{\Theta}[L_f](\theta) = \frac{1}{\sum_{O_j \in L_f} P_j^L} \sum_{O_j \in L} P_j^L . m_j^{\Theta}(\theta) \quad \theta \subseteq \Theta \tag{28}$$

where $P_j^{L_f}$ is the probability of the instance O_j to belong to the leaf L_f (i.e. $f \in [1, F]$). This latter is computed as the cross product of the pignistic probabilities of the object O_j to belong to the nodes that link the root node and the corresponding leaf node L_f.

Example 3. Assuming Example 2, we try to generate our GR-BDT classifier. As it is shown in the aforementioned example, we have deduced the performance of the $Income$ attribute over the $Property$ and the $UnpaidCredit$ attributes. Accordingly, the $Income$ attribute has been chosen as a root node. The first generated tree is given in Figure 1.

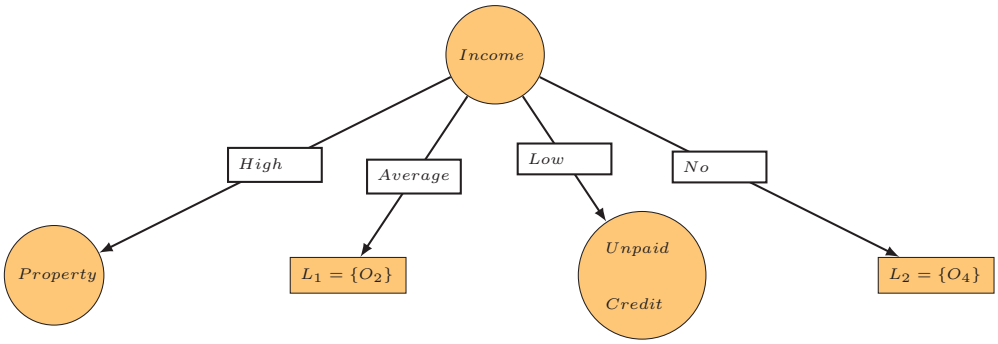

Figure 1: The first generated tree of the GR-BDT classifier

We notice that both subsets $S^{Income}_{Average}$ and S^{Income}_{No} contain only one object and accordingly they have fulfilled one of the stopping criteria and they have declared as leaves having respectively as bbas $m^{\Theta}[L_1]$ and $m^{\Theta}[L_2]$. Nonetheless, the subsets S^{Income}_{High} and S^{Income}_{No} have not satisfied the stopping criteria. So, we apply the same process for the both subsets until the stopping criteria hold. The final generated decision tree is depicted in Figure 2.

6.2. DiffRatio Belief Decision tree (DR-BDT)

We provide, in the following, the parameters enabling the construction of the DR-BDT classifier.

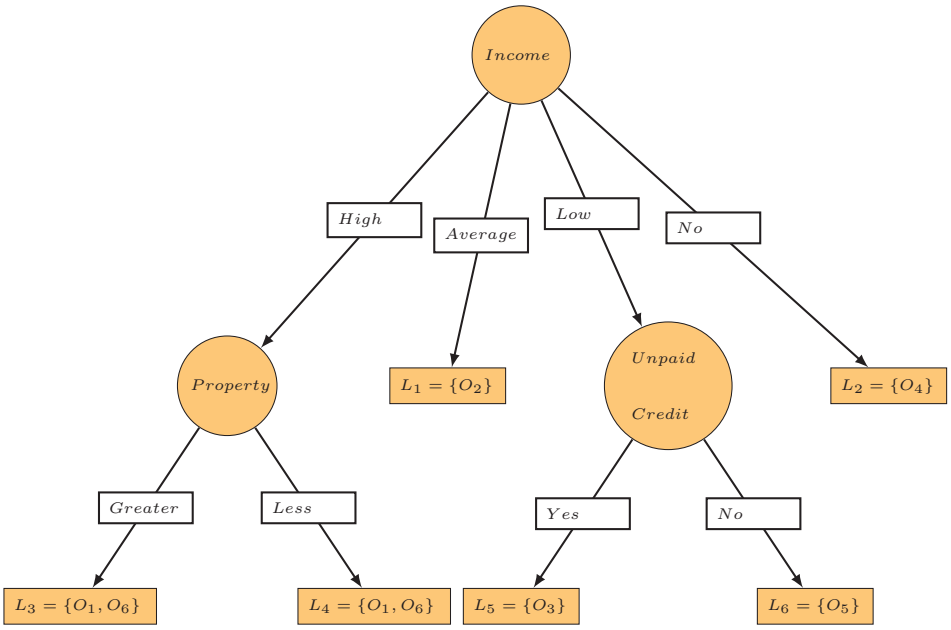

Figure 2: The final generated tree of the *GR*-BDT classifier

6.2.1. Attribute selection measure

By analogy with [17], the *DR*-BDT classifier consists of computing the intra-group distance that measures for each attribute value how much objects are close to each other. We propose the following steps to pick out the best decision attribute:

1. We compute the intra-group distance between objects of S in terms of their classes:

$$SumD(S) = \sum_{O_i \in S} \sum_{O_{j \geq i+1} \in S} dist(m_i^{\Theta}, m_j^{\Theta}) \tag{29}$$

where *dist* represents the Jousselme distance between the two bbas m_i^{Θ} and m_j^{Θ} that correspond respectively to the bbas relative to the class of

objects O_i and O_j.

2. Then, for each attribute value v, we compute $SumD(S_v^{A_k})$ as follows:

$$SumD(S_v^{A_k}) = \sum_{O_i \in S} \sum_{O_{j \geq i+1} \in S} D_i^{S_v^{A_k}} . D_j^{S_v^{A_k}} . dist(m_i^{\Theta}, m_j^{\Theta}) \tag{30}$$

where $D_i^{S_v^{A_k}}$ reflects the belonging of the object O_i to the subset $S_v^{A_k}$ and it is set to:

$$D_i^{S_v^{A_k}} = 1 - dist(m^{\Theta^{A_v^k}}, m_i^{\Theta^{A_k}}) \tag{31}$$

$m^{\Theta^{A_v^k}}$ is a certain bba where $m^{\Theta^{A_v^k}}(v) = 1$.

3. For each attribute $A_k \in A$, we compute $SumD_{A_k}(S)$ as follows:

$$SumD_{A_k}(S) = \sum_{v \in \Theta^{A_k}} SumD(S_v^{A_k}). \tag{32}$$

4. We compute the $DiffRatio$ relative to the attribute A_k, where the $SplitInfo$ will be calculated such as in Equation 14.

$$DiffRatio(S, A_k) = \frac{SumD(S) - SumD_{A_k}(S)}{SplitInfo(S, A_k)}. \tag{33}$$

Drawing inspiration from the clustering approach, the idea underlying the $DiffRatio$ approach consists of minimizing the intra distance between objects, while maximizing the inter distance. That is the $DiffRatio$ measure allows to band together similar objects as much as possible.

5. We repeat this process for each attribute $A_k \in A$ and then select the one that maximize the $DiffRatio$.

Example 4. Let us continue with the training data given in Example 1 and try to illustrate the attribute selection process when S=T on the basis of the $DiffRatio$ criterion.

- We start by calculating the intra-group distance between objects in terms of their classes using Equation 29. We get:
  - $SumD(T)$= 9.0741.
- Then, we compute the total distance induced by each attribute. We obtain:
  - $SumD_{Income}(T) = 0.7479$,
  - $SumD_{Property}(T) = 3.5076$,
  - $SumD_{UnpaidCredit}(T) = 3.8291$.
- Then, we compute the $DiffRatio$ for each attribute:
  - $DiffRatio(Income) = 4.3404$,
  - $DiffRatio(Property) = 5.5677$,
  - $DiffRatio(UnpaidCredit) = 5.2450$.

  From this, we can deduce that the attribute $Property$ is the one that maximises the $DiffRatio$ criterion. Accordingly, it will be chosen as the root node of the tree.

6.2.2. Partitioning strategy

Each training object O_i, within the DR-BDT classifier, may be part of more than one subset $S_v^{A_k}$ with a degree of belonging $D_i^{S_v^{A_k}}$:

- When $D_i^{S_v^{A_k}} = 1$, the object O_i belongs to the branch having v as a value.
- When $D_i^{S_v^{A_k}} = 0$, the object O_i may not belong to the branch having v as label. In contrast, it will be part of the remaining branches.
- else, the object O_i belongs to all the branch of the attribute A_k

6.2.3. Stopping criteria

The stopping criteria relative to the DR-BDT approach are similar to those of the GR-BDT:

1. Only one instance is part of the treated node.
2. Instances of the treated node belong to the same class.
3. There is no further attribute for checking.
4. The remaining attributes have $DiffRatio$ equal or less than zero.

6.2.4. Structure of leaves

Leaves L_f, within this approach, may contain objects O_i with different class labels $\theta \in \Theta$. Thus, each leaf L_f has to be characterized by a belief function $m^{\Theta}[L_f]$ that is calculated in terms of the used approach.

Let $L_f = v_f^1 \times \ldots \times v_f^J \in A_f^1 \times \ldots \times A_f^J$ be the set of attribute values that correspond to the branch leading to L_f ($f \in [1, F]$) (i.e J corresponds to the

number of edges from the tree's root node to the leaf node). The leaf mass $m^{\Theta}[L_f]$ will then be computed as:

$$m^{\Theta}[L_f] = \textcircled{\cap}_{O_i \in L_f} \hat{m_i}^{\Theta} \tag{34}$$

where $\hat{m_i}^{\Theta}$ reflects the discounted bba relative to m_i^{Θ} and is set to:

$$\hat{m_i}^{\Theta}(\theta) = (1 - \alpha_f).m_i^{\Theta}(\theta) \quad \theta \subset \Theta \tag{35}$$

$$\hat{m_i}^{\Theta}(\Theta) = \alpha_f + (1 - \alpha_f).m_i^{\Theta}(\Theta) \tag{36}$$

with

$$\alpha_f = \frac{1}{J}.\sum_{t=1}^{J}(1 - D_i^{S_{v_f^t}^{A_f^t}}). \tag{37}$$

Example 5. We move on now to generate the *DR*-BDT classifier relative to the training data given in Table 1. According to Example 4, we notice that the attribute *Property* has outperformed both the *UnpaidCredit* and the *Income* attributes. Consequently, it should be chosen as the decision tree root. The first generated tree is given in Figure 3.

Both subsets $S_{Greater}^{Property} = \{O_1, O_2, O_3, O_5, O_6\}$ and $S_{Less}^{Property} = \{O_1, O_3, O_4, O_5, O_6\}$ have not fulfilling the stopping criteria. Therefore, the same process has to be applied for these subsets until a stopping criterion is satisfied. The final generated decision tree is given in Figure 4.

7. Experimentation settings and results

In this section, we evaluate the performance of our proposed decision tree approaches. In the following, we detail our experimentation settings and results.

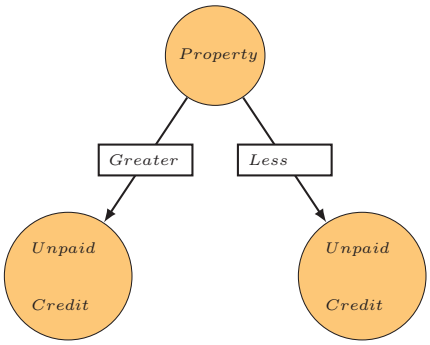

Figure 3: The first generated tree of the *DR*-BDT classifier

7.1. Experimentation settings

With the aim of evaluating the performance of our three evidential classifiers, we made experiments on several real world databases acquired from the UCI machine learning databases [32]. We have also picked out several databases that have missing values. Table 4 provides the details of the used databases, where #Instances, #Attributes, #Classes and #Missing denote, respectively, the number of instances, the number of attributes, the number of classes and the existence or not of missing values.

Table 4: Description of databases

Databases	#Instances	#Attributes	#Classes	#Missing
Breast Cancer	286	9	2	Yes
Primary Tumor	339	17	22	Yes
Voting Records	435	16	2	Yes
Soybean	638	36	2	Yes
Ecoli	336	8	8	No
Hepatitis	155	19	2	Yes
Auto MPG	398	8	3	Yes
Wine	178	13	3	No
Echo-cardiogram	132	12	2	Yes

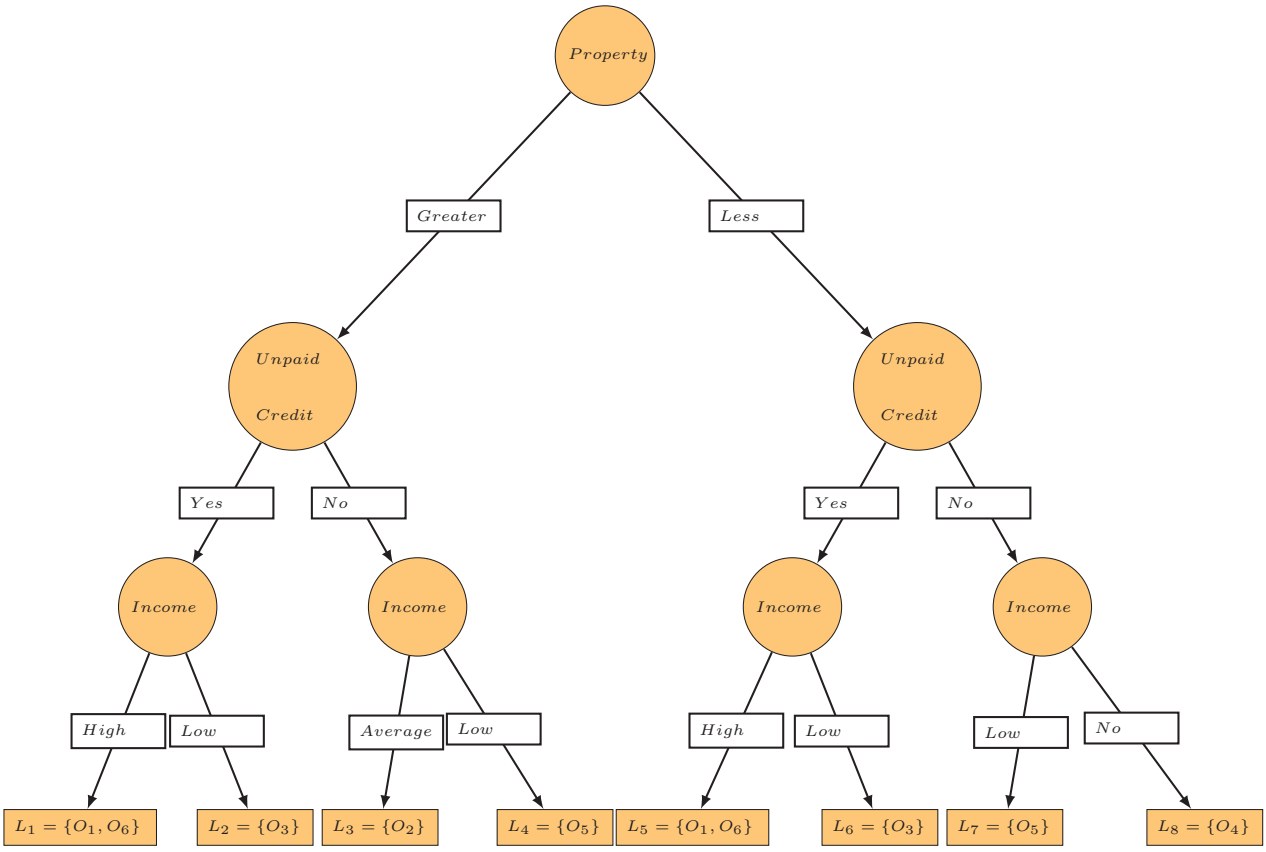

Figure 4: The final generated tree of the DR-BDT classifier

From a practical point of view, missing values have to be imputed and continuous variables have usually to be discretized into bins. However, the uncertainty introduced by missing values imputation and continuous variables discretization have to be addressed. In this paper, we propose to express data uncertainty with the belief function theory. From this, we assign certain bbas to categorical attribute values and we transform the continuous values into beliefs using the Evidential c-Means approach (ECM) [33, 34]. Regarding

the missing data, we propose to impute these data using the Nearest Neighbors imputation approach [35]. As we handle evidential data, we used the Jousselme distance measure instead of the standard distance metrics for computing the distance between instances. That is to say, the distance between the object with missing values and the objects with complete values should be calculated. The k Nearest Neighbors objects is then used for the imputation process. So, we assign, for each object with a missing attribute value, the mean bba of its k-Nearest Neighbors corresponding to that attribute.
Two kinds of experimentation have been considered. The first one aims to evaluate the performance of our decision tree versions against standard decision tree classifiers, particularly the C4.5 algorithm. Our second experimentation concerns the evaluation and the comparison of our decision tree approaches against existing evidential classifiers. For the comparison process, we have run the 5-folds cross validation technique and we have relied on the Percentage of Correctly Classification measure (PCC) as an evaluation criterion.

7.2. Experimentation results

This Section is devoted to comparing our proposed decision tree approaches over standard and evidential classifiers. Before tackling the comparison process, we suggest to compare both the $GR{-}BDT$ and the $DR{-}BDT$ in the way of selecting decision attributes. That is to say, we relied on the Spearman's rank correlation coefficient [36] for measuring the rank correlation between both approaches achieved at the root node. This rank measure takes values in the range between -1 and 1. A value of -1, 0 and 1 state respectively

the case of negative correlation (different and dependent results), no correlation (different and independent results) and positive correlation (similar results) between the GR-BDT and the DR-BDT. Table 5 provides the rank correlation results for the different databases, where ICM corresponds to the Index Correlation Measure.

Table 5: Rank correlation approach between the GR-BDT and the DR-BDT

Databases	ICM
Breast Cancer	0.11
Primary Tumor	-0.67
Voting Records	0.32
Soybean	0
Ecoli	-0.89
Hepatitis	0.09
Auto MPG	-0.4
Wine	-0.15
Echo-cardiogram	-0.73

The results obtained in Table 5 indicate the high difference between the GR-BDT and the DR-BDT when selecting attributes. In fact, almost all databases have yielded index correlation measure in the range [0,-1]. Note that, we compare both the GR-BDT and the DR-BDT against the standard C4.5 algorithm and some evidential decision tree classifiers.

7.2.1. Experimentation1: Comparison against the C4.5 algorithm

In this section, we experiment our proposed approaches and we compare them to the C4.5 algorithm using evidential databases. To do so, we transform categorical datasets with missing data upon the use of the evidential database transformation process. The comparison results in terms of the

PCC criterion is given in Table 6. From this table, we can remark that the mean PCC yielded by the *GR*-BDT (84.68%) and the *DR*-BDT (85.61 %) has outperformed that obtained using the C4.5 algorithm (75.53%). From this, we can deduce that both our approaches yield interesting results and we can conclude that our proposed decision tree approaches have outperformed the C4.5 algorithm.

Table 6: Comparison of percentage of correctly classification against the C4.5 algorithm (%)

	C4.5	*GR*-BDT	*DR*-BDT
Breast Cancer	74.12 ± 0.44	**78.14 ± 0.17**	76.52 ± 0.23
Primary Tumor	40.70 ± 0.19	69.23 ± 0.40	**72.43 ± 0.27**
Voting Records	96.55 ± 0.17	97.01 ± 0.14	**97.83 ± 0.36**
Soybean	90.77 ± 0.09	94.35 ± 0.20	**95.67 ± 0.15**
Mean	75.53	**84.68**	**85.61**

7.2.2. Experimentation 2: Comparison with the naive and the E2M classifiers

This experimentation concerns the comparison of our decision tree approaches against the E2M DT and the naive approach presented in [37]. The experimentation results are presented in Table 7, where we can remark that the mean PCCs achieved by the *GR*-BDT and the *DR*-BDT are a little bit greater then that obtained by the E2M DT and the Naive classifier. From this, we can deduce the efficiency of our proposed decision tree algorithms against the Naive and the E2M DT classifiers.

Table 7: Comparison of percentage of correctly classification against the naive and the E2M algorithms (%)

	Naive	E2M DT	*GR*-BDT	*DR*-BDT
Breast Cancer	74.63 ± 0.13	**79.51 ± 0.27**	78.14 ± 0.17	76.52 ± 0.23
Primary Tumor	63.17 ± 0.33	68.76 ± 0.05	69.23 ± 0.40	**72.43 ± 0.27**
Voting Records	81.89 ± 0.27	95.65 ± 0.14	97.01 ± 0.14	**97.83 ± 0.36**
Soybean	89.23 ± 0.24	93.86 ± 0.32	94.35 ± 0.20	**95.67 ± 0.15**
Ecoli	83.56 ± 0.02	**85.11 ± 0.17**	84.26 ± 0.09	84.92 ±0.21
Hepatitis	78.05 ± 0.36	83.55 ± 0.31	**85.37 ± 0.19**	84.79 ± 0.35
Auto MPG	80.60 ± 0.07	85.23 ± 0.29	**93.18 ± 0.15**	92.56 ± 0.26
Wine	72.33 ± 0.37	75.14 ± 0.11	78.32 ± 0.32	**80.47 ± 0.08**
Echo-cardiogram	94.56 ± 0.43	**97.14 ± 0.15**	96.26 ± 0.34	95.69 ± 0.16
Mean	79.78	84.88	**86.23**	**86.76**

8. Conclusion

In this paper, we have proposed two decision tree classifier techniques to cope with imperfect knowledge expressed within the evidence theory. Precisely, we have tackled data with uncertain attribute values as well as uncertain class labels. Regarding the future research directions, we look forward to investigating more robust techniques for uncertain data modeling within the belief function framework. Notably, the case of incomplete data should be well studied and an extension of the E2M algorithm in the context of evidential data may be quite sufficient. We intend also to apply a pruning strategy with the aim of improving predictive accuracies. It would also be interesting to develop ensemble of our *GR*-BDT and *DR*-BDT. More concretely, we look forward to constructing *GR*-BDT and *DR*-BDT random forests.

9. Acknowledgment

We would like to express our deep gratitude to Thierry Denoeux, Sébastien Destercke and Nicolas Sutton Charani for their help in doing the comparative study between our two proposed approaches and their E2M DT classifier. Thanks are also due to the anonymous reviewers for their constructive and helpful comments and suggestions which have been in help to improve the quality of this paper.

References

[1] Z. Elouedi, K. Mellouli, P. Smets, Classification with belief decision trees, in: Artificial Intelligence: Methodology, Systems, and Applications, Springer, 2000, pp. 80–90.

[2] J. R. Quinlan, Decision trees as probabilistic classifiers, in: the 4th International Workshop on Machine Learning, 1987, pp. 31–37.

[3] M.-H. Masson, T. Denœux, Ranking from pairwise comparisons in the belief functions framework, in: Belief Functions: Theory and Applications, Springer, 2012, pp. 311–318.

[4] M. Umanol, H. Okamoto, I. Hatono, H. Tamura, F. Kawachi, S. Umedzu, J. Kinoshita, Fuzzy decision trees by fuzzy ID3 algorithm and its application to diagnosis systems, in: Proceedings of the Third IEEE World Congress Conference on Computational Intelligence., IEEE, 1994, pp. 2113–2118.

[5] E. Hüllermeier, Possibilistic induction in decision-tree learning, in: European Conference on Machine Learning, Springer, 2002, pp. 173–184.

[6] D. Dubois, H. Prade, Decision evaluation methods under uncertainty and imprecision, in: Combining fuzzy imprecision with probabilistic uncertainty in decision making, Springer, 1988, pp. 48–65.

[7] J. Bezdek, et al., Fuzziness vs. probability-again (!?), IEEE Transactions on Fuzzy systems 2 (1) (1994) 1–3.

[8] B. Qin, Y. Xia, F. Li, DTU: a decision tree for uncertain data, Advances in Knowledge Discovery and Data Mining (2009) 4–15.

[9] T. Denœux, Reasoning with imprecise belief structures, International Journal of Approximate Reasoning 20 (1) (1999) 79–111.

[10] Z. Elouedi, K. Mellouli, P. Smets, Belief decision trees: theoretical foundations, International Journal of Approximate Reasoning 28 (2) (2001) 91–124.

[11] S. Trabelsi, Z. Elouedi, K. Mellouli, Pruning belief decision tree methods in averaging and conjunctive approaches, International Journal of Approximate Reasoning 46 (3) (2007) 568–595.

[12] P. Vannoorenberghe, T. Denoeux, Handling uncertain labels in multiclass problems using belief decision trees, in: Proceedings of International Conference on Information Processing and Management of Uncertainty, Vol. 3, 2002, pp. 1919–1926.

[13] N. Sutton-Charani, S. Destercke, T. Denœux, Classification trees based on belief functions, in: Belief Functions: Theory and Applications, Springer, 2012, pp. 77–84.

[14] N. Sutton-Charani, S. Destercke, T. Denœux, Training and evaluating classifiers from evidential data: Application to E2M decision tree pruning, in: Proceedings of International Conference on Belief Functions, Springer, 2014, pp. 87–94.

[15] L. Ma, S. Destercke, Y. Wang, Online active learning of decision trees with evidential data, Pattern Recognition 52 (2016) 33–45.

[16] A. Trabelsi, Z. Elouedi, E. Lefevre, Handling uncertain attribute values in decision tree classifier using the belief function theory, in: Proceedings of International Conference on Artificial Intelligence: Methodology, Systems, and Applications, Springer, 2016, pp. 26–35.

[17] A. Trabelsi, Z. Elouedi, E. Lefevre, New decision tree classifier for dealing with partially uncertain data, in: Proceedings of 25ème Rencontres francophones sur la Logique Floue et ses Applications, 2016, pp. 57–64.

[18] P. Smets, The combination of Evidence in the Transferable Belief Model, IEEE Transactions on Pattern Analysis and Machine Intelligence 12 (5) (1990) 447–458.

[19] G. Shafer, A mathematical theory of evidence, Vol. 1, Princeton university press Princeton, 1976.

[20] P. Smets, The application of the Transferable Belief Model to diagnostic

problems, International Journal of Intelligent Systems 13 (1998) 127–157.

[21] P. Smets, The transferable belief model for expert judgements and reliability problems, Reliability Engineering & System Safety 38 (1-2) (1992) 59–66.

[22] P. Smets, The Transferable Belief Model for quantified belief representation, In Handbook of Defeasible Reasoning and Uncertainty Management Systems 1 (1988) 267–301.

[23] A. Jousselme, D. Grenier, E. Bossé, A new distance between two bodies of evidence, Information Fusion 2 (2) (2001) 91–101.

[24] B. Tessem, Approximations for efficient computation in the theory of evidence, Artificial Intelligence 61 (2) (1993) 315–329.

[25] L. Breiman, J. H. Friedman, R. A. Olshen, C. J. Stone, Classification and Regression Trees, Wadsworth, 1984.

[26] J. R. Quinlan, Induction of decision trees, Machine learning 1 (1) (1986) 81–106.

[27] J. R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann, 1993.

[28] R. L. De Mántaras, A distance-based attribute selection measure for decision tree induction, Machine learning 6 (1) (1991) 81–92.

[29] T. Denœux, Conjunctive and disjunctive combination of belief functions

induced by nondistinct bodies of evidence, Artificial Intelligence 172 (2) (2008) 234–264. doi:10.1016/j.artint.2007.05.008.

[30] P. Smets, Belief functions: the disjunctive rule of combination and the generalized bayesian theorem, in: Classic Works of the Dempster-Shafer Theory of Belief Functions, Springer, 2008, pp. 633–664.

[31] G. J. Klir, D. Harmanec, Types and measures of uncertainty, in: Consensus under fuzziness, Springer, 1997, pp. 29–51.

[32] P. Murphy, D. Aha, UCI repository databases, http://www.ics.uci.edu/mlear.

[33] M.-H. Masson, T. Denoeux, Ecm: An evidential version of the fuzzy c-means algorithm, Pattern Recognition 41 (4) (2008) 1384–1397.

[34] A. Samet, E. Lefèvre, S. Ben Yahia, Evidential data mining: precise support and confidence, Journal of Intelligent Information Systems 47 (1) (2016) 135–163.

[35] N. L. Crookston, A. O. Finley, yaimpute: an R package for knn imputation, Journal of Statistical Software 23 (10) (2008) 16.

[36] C. Spearman, The proof and measurement of association between two things, The American journal of psychology 15 (1) (1904) 72–101.

[37] N. Sutton-Charani, S. Destercke, T. Denœux, Learning decision trees from uncertain data with an evidential em approach, in: Proceeding of the 12th International Conference on Machine Learning and Applications, Vol. 1, IEEE, 2013, pp. 111–116.