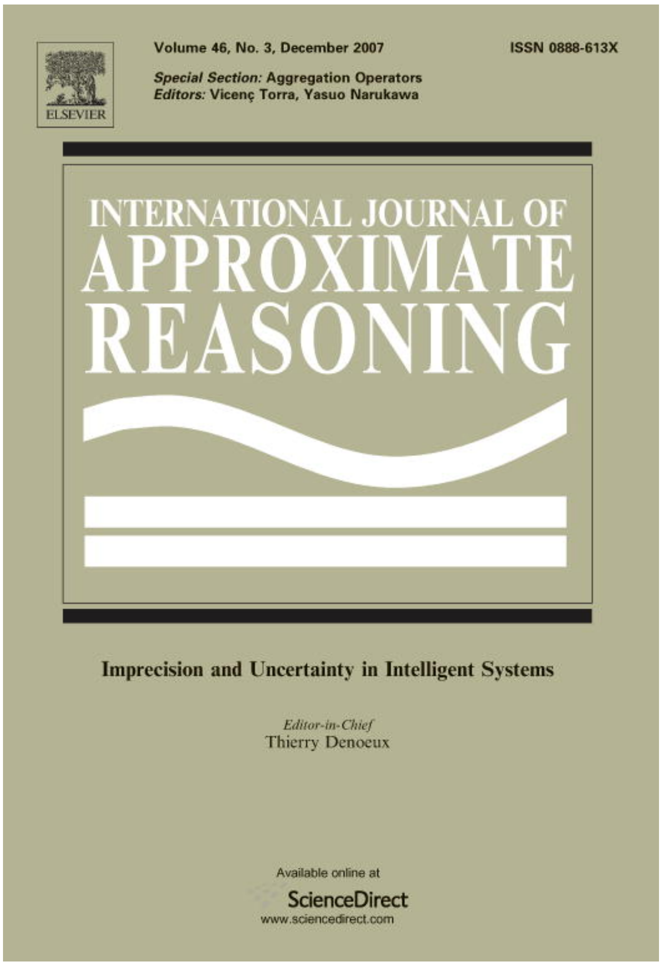
Volume 46, No. 3, December 2007
ISSN 0888-613X
ELSEVIER
Special Section: Aggregation Operators
Editors: Vicenç Torra, Yasuo Narukawa
INTERNATIONAL JOURNAL OF
APPROXIMATE
REASONING
Imprecision and Uncertainty in Intelligent Systems
Editor-in-Chief
Thierry Denoeux
Available online at
ScienceDirect
www.sciencedirect.com

Available online at www.sciencedirect.com
ScienceDirect
International Journal of Approximate Reasoning
46 (2007) 568–595
INTERNATIONAL JOURNAL OF APPROXIMATE REASONING
www.elsevier.com/locate/ijar

Pruning belief decision tree methods in averaging and conjunctive approaches

Salsabil Trabelsi, Zied Elouedi *, Khaled Mellouli

LARODEC, Institut Supérieur de Gestion de Tunis, 41 Avenue de la Liberté, 2000 Le Bardo, Tunis, Tunisia

Received 1 August 2006; received in revised form 31 January 2007; accepted 7 February 2007
Available online 4 March 2007

Abstract

The belief decision tree (BDT) approach is a decision tree in an uncertain environment where the uncertainty is represented through the Transferable Belief Model (TBM), one interpretation of the belief function theory. The uncertainty can appear either in the actual class of training objects or attribute values of objects to classify. From the procedures of building BDT, we mention the averaging and the conjunctive approaches.

In this paper, we develop pruning methods of belief decision trees induced within averaging and conjunctive approaches where the objective is to cope with the problem of overfitting the data in BDT in order to improve its comprehension and to increase its quality of the classification.

Keywords: Uncertainty; Belief function theory; Decision tree; Belief decision tree; Pruning

1. Introduction

In recent years, many learning techniques are used to ensure classification. Among them, decision trees are considered as one of the efficient classification techniques applied successfully in many areas such as expert systems, medical diagnoses, speech recognition, etc. They can also be used in other fields like marketing, finance, industry, etc. Decision trees have become popular in the area of machine learning.

* Corresponding author. Tel.: +216 98354307.
E-mail addresses: Salsabilt@voila.fr (S. Trabelsi), zied.elouedi@gmx.fr (Z. Elouedi), khaled.mellouli@ihec.rnu.tn (K. Mellouli).

0888-613X/$ - see front matter © 2007 Elsevier Inc. All rights reserved.
doi:10.1016/j.ijar.2007.02.004

However, standard decision trees do not well perform their classification task in an environment characterized by uncertainty in data. In order to overcome this limitation, many researches have been done to adapt standard decision trees to this kind of environment. The idea was to introduce theories that could represent uncertainty. Several kinds of decision trees were developed: probabilistic decision trees [19,20], fuzzy decision trees [32,28], belief decision trees [6,9,30,31] and possibilistic decision trees [2,13,14]. In our work, we will focus on belief decision trees developed in [9].

The belief decision tree approach is a decision tree technique adapted in order to handle uncertainty about the actual class of the objects in the training set and also to classify objects characterized by uncertain attributes. The uncertainty is represented by the Transferable Belief Model (TBM), one interpretation of the belief function theory. It is considered as a useful theory for representing and managing uncertain knowledge introduced by [4,22]. It allows to express partial or total ignorance in a flexible way. The belief function theory is appropriate to handle uncertainty in classification problems especially within the decision tree technique. So, BDT is based on both the decision tree technique and belief function theory to handle uncertainty.

When a belief decision tree is built within the averaging or conjunctive building procedures from real world databases, many of branches will reflect noise in the training data due to uncertainty. The results are many of undesirable nodes and difficulty to interpret the tree. Our aim is to overcome this problem of overfitting in belief decision tree. In order to reduce the size and to improve the classification accuracy.

Pruning is a way to cope with this problem. So, our objective in this work is to prune belief decision tree. ''How does tree pruning work?'' there are two common approaches to tree pruning. Methods that can control the growth of a decision tree during its development are called *pre-pruning* methods, the others are called post-pruning methods. In post-pruning approach, grow the full tree, allow it overfit the data and then post-prune it. It requires more computation than pre-pruning, yet generally leads to a more reliable tree. So, the beneficial effects of a post-pruning approach have attracted the attention of many researches, who proposed a number of methods like minimal cost-complexity pruning [3], reduced error pruning [18], critical value pruning [15], pessimistic error pruning [18], minimum error pruning [16] and error based pruning [21]. However, all these methods deal with only standard decision trees and not with BDT.

In this work, we focus on post-pruning approach to simplify the BDT. Pre-pruning has been developed in [10] by improving the stopping criteria concerning the value of the selection measure in BDT using a discounting factor which depends on the number of objects in a node and in [6] where impurity measure, based on evidence-theoretic uncertainty measure, is used to grow the tree and has the advantage to define simultaneously the pruning strategy. It allows to control the complexity of the tree, thus avoiding overtraining. This latter has been extended to multiclass problems in [30]. So, we suggest to develop post-pruning methods based on one method of post-pruning approach to prune the BDT in averaging and conjunctive approaches introduced in [16] in order to reduce the complexity and to increase the quality of classification.

This paper is organized as follows: Section 2 presents the basic concepts of decision trees, Section 3 provides a brief description of basics of belief function theory as explained by the TBM. In section 4, we describe the BDT within averaging and conjunctive approaches. Then, in Section 5, we present the description of our pruning belief decision tree methods in both approaches. Finally in Section 6, we carry simulations to compare BDT without pruning, after pre-pruning introduced in [10] and after our post-pruning methods.

2. Decision trees

A decision tree is a flow-chart-like tree structure allowing to determine the class of an object given known values of its attributes. The visual presentation makes the decision tree model very easy to understand. 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 subdecision tree.
3. *A leaf* which is also named an answer node, including objects that, typically, belong 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:

- *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.
- *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 remove down the tree branch according to the attribute value of the given instance. This process is repeated until a leaf is encountered and which is characterized by a class.

Several algorithms for building decision trees have been developed such as *ID3* and its successor *C4.5* algorithm [17,21]. We can also mention the *CART* algorithm [3]. In addition to these non-incremental algorithms, many incremental building decision trees algorithms have been proposed such as *ID5* [29].

The formalism for building decision trees is referred to 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 parameters:

1. *The attribute selection measure:* A critical parameter, generally based on the information theory [23], for building decision trees. It allows determining the best attribute for each node, in order to partition the training set T. We can for instance mention those suggested by Quinlan: *the information gain* [17] and *the gain ratio* [21].
2. *The partitioning strategy:* The current training set will be divided by taking into account the selected test attribute. In the case of symbolic attributes, this strategy consists in testing all the possible attributes values, whereas in the case of numeric attributes, a discretization step is generally needed [12].
3. *The stopping criteria:* Generally, we stop the partitioning process if one of the following situations is presented:

- All the remaining objects belong to only one class.
- No further attribute to test.
- All the remaining attributes have no informational contribution.

The choice of these parameters makes the major difference between decision tree algorithms.

2.1. The C4.5 algorithm

C4.5 is a supervised learning algorithm developed by Quinlan as successor of ID3 algorithm. The C4.5 algorithm uses *the gain ratio criterion* as the attribute selection measure. This parameter is expressed as follows:

$$Gain\ ratio(T,X) = \frac{Gain(T,X)}{Split\ Info(T,X)} \tag{1}$$

$$Gain(T,X) = Info(T) - Info_X(T) \tag{2}$$

$$\text{Where } Info(T) = -\sum_{i=1}^{n} \frac{freq(C_i,T)}{|T|} \log_2 \frac{freq(C_i,T)}{|T|} \tag{3}$$

$$\text{and } Info_X(T) = -\sum_{v \in D(X)} \frac{|T_v^X|}{|T|} Info(T_v^X) \tag{4}$$

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

where T is a training set and each instance in T belongs to one class from the set of n mutually exclusive and exhaustive classes. $Info(T)$ represents the entropy of a set T of instances. $freq(C_i,T)$ denotes the number of objects in the training set T belonging to the class C_i. T_v^X denotes the training subset of T containing objects whose attribute X has v as value and $D(X)$ is the domain of the possible values of the attribute X. $Gain(T,X)$ is defined as the expected reduction in entropy resulting from partitioning T according to X. $SplitInfo(T,X)$ represents the potential information generated by dividing T into $|D(X)|$ subsets. So, $Gainratio(T,X)$ expresses the proportion of information generated by the split that appears helpful for classification.

The gain ratio of each attribute is computed and the one presenting the highest value will be selected.

The different steps of the C4.5 algorithm are summarized as follows:

1. If all instances belong to only one class, then the decision tree is a leaf labeled with that class.
2. Otherwise,
   - Select the attribute that maximizes $Gainratio(T,X)$.
   - Split the training set T into several subsets, one for each value of the selected attribute.
   - Apply the same procedure recursively for each generated subset.

3. Belief function theory

In this section, we briefly review the main concepts underlying the belief function theory as interpreted by the Transferable Belief Model (TBM). This theory is appropriate to handle uncertainty in classification problems especially within the decision tree technique.

3.1. Definitions

The TBM is a model to represent quantified belief functions [25]. Let Θ be a finite set of elementary events to a given problem, called the frame of discernment [24]. All the subsets of Θ belong to the power set of Θ, denoted by 2^{Θ}.

The impact of a piece of evidence on the different subsets of the frame of discernment Θ is represented by a basic belief assignment (bba).

The bba is a function $m : 2^{\Theta} \to [0,1]$ such that:

$$\sum_{A \subseteq \Theta} m(A) = 1 \tag{6}$$

The value $m(A)$, named a basic belief mass (bbm), represents the portion of belief committed exactly to the event A.

Associated with m is the belief function, denoted *bel*, corresponding to a specific bba m, assigns to every subset A of Θ the sum of masses of belief committed to every subset of A by m [22]. Contrary to the bba which expresses only the part of belief that one commits to A without being also committed to $\overline{A}$.

The belief function *bel* is defined for $A \subseteq \Theta, A \neq \emptyset$ as

$$bel(A) = \sum_{\emptyset \neq B \subseteq A} m(B) \tag{7}$$

The plausibility function *pl* quantifies the maximum amount of belief that could be given to a subset A of the frame of discernment. It is equal to the sum of the bbm's relative to subsets B compatible with A.

The plausibility function *pl* is defined as follows:

$$pl(A) = \sum_{A \cap B \neq \emptyset} m(B) \quad \forall A \subseteq \Theta \tag{8}$$

Another function is used basically to simplify computations namely the commonality function q [1] is defined as follows:

$$q(A) = \sum_{A \subseteq B} m(B) \quad \forall A \subseteq \Theta \tag{9}$$

The basic belief assignment (m), the belief function (*bel*), the plausibility function (*pl*) and the commonality function (q) are considered as different expressions of the same information [5].

3.2. Combination

Handling information induced from different experts (information sources) requires an evidence gathering process in order to get the fused information. In the transferable belief model, the basic belief assignments induced from distinct pieces of evidence are combined by either the conjunctive rule or the disjunctive rule of combination.

1. The conjunctive rule: When we know that both sources of information are fully reliable then the bba representing the combined evidence satisfies [26]:

$$(m_1 \textcircled{\cap} m_2)(A) = \sum_{B,C \subseteq \Theta : B \cap C = A} m_1(B) m_2(C) \tag{10}$$

This rule can be simply computed in terms of the commonality functions as follows:

$$(q_1 \textcircled{\cap} q_2)(A) = q_1(A) q_2(A) \tag{11}$$

where q_1 and q_2 are respectively the commonality functions corresponding respectively to the bba's m_1 and m_2.

2. The disjunctive rule: When we only know that at least one of sources of information is reliable but we do not know which is reliable, then the bba representing the combined evidence satisfies [26]:

$$(m_1 \textcircled{\cup} m_2)(A) = \sum_{B,C \subseteq \Theta : B \cup C = A} m_1(B) m_2(C) \tag{12}$$

3.3. Discounting

In the transferable belief model, discounting allows to take in consideration the reliability of the information source that generates the bba m. For $\alpha \in [0,1]$, let $(1-\alpha)$ be the degree of confidence ('reliability'), we assign to the source of information. If the source is not fully reliable, the bba it generates is 'discounted' into a new less informative bba denoted m^{α}:

$$m^{\alpha}(A) = (1-\alpha) m(A) \quad \text{for } A \subset \Theta \tag{13}$$

$$m^{\alpha}(\Theta) = \alpha + (1-\alpha) m(\Theta) \tag{14}$$

3.4. Decision making

In the transferable belief model, holding beliefs and making decisions are distinct processes. Hence, it proposes two level models:

- *The credal level* where beliefs are represented by belief functions.
- *The pignistic level* where beliefs are used to make decisions and represented by probability functions called the pigninstic probabilities denoted BetP [27] and is defined as

$$BetP(A) = \sum_{B \subseteq \Theta} \frac{|A \cap B|}{|B|} \frac{m(B)}{(1 - m(\emptyset))} \quad \text{for all } A \in \Theta \tag{15}$$

4. Belief decision trees

A belief decision tree is a decision tree in an uncertain environment where the uncertainty is represented by the TBM. In our work, we focus on BDT proposed in [9] where there are two approaches of building. These latters deal with only symbolic attributes.

- The averaging approach [7] is an extension of the classical approach developed by Quinlan and based on the gain ratio criterion [21].
- The conjunctive approach [8] represented ideas behind the TBM itself and based on a distance criterion.

We will use the following notations:

- T: a given training set composed by p objects I_j, $j = 1, \ldots p$,
- S: a set of objects belonging to the training set T,
- X: an attribute,
- $\Theta = \{C_1, C_2, \ldots, C_n\}$: the frame of discernment made of the n possible classes related to the classification problem.
- $m^{\Theta}\{I_j\}(C)$: the bbm given to the hypothesis that the actual class of object I_j belongs to $C \subseteq \Theta$.

4.1. Belief decision tree parameters

In this section, we define the major parameters leading to the construction of the belief decision tree where objects may have uncertain classes.

4.1.1. The attribute selection measure

The major parameter ensuring the building of a decision tree is the attribute selection measure allowing to determine the attribute to assign to a node of the induced BDT at each step.

4.1.1.1. Averaging approach. Under this approach, the attribute selection measure is based on the entropy computed from the average pignistic probabilities computed from the pignistic probabilities of each instance in the node. The following steps are proposed to choose the appropriate attribute:

1. Compute the pignistic probability of each object I_j by applying the pignistic transformation to $m^{\Theta}\{I_j\}$.
2. Compute the average pignistic probability function $BetP^{\Theta}\{S\}$ taken over the set of objects S. For each $C_i \in \Theta$,

$$BetP^{\Theta}\{S\}(C_i) = \frac{1}{|S|} \sum_{I_j \in S} BetP^{\Theta}\{I_j\}(C_i) \tag{16}$$

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

$$Info(S) = -\sum_{i=1}^{n} BetP^{\Theta}\{S\}(C_i) \log_2 BetP^{\Theta}\{S\}(C_i) \tag{17}$$

4. Select an attribute X. Collect the subset S_v^X made with cases of S having v as a value for the attribute X. Then, compute the average pignistic probability for objects in subset S_v^X. Let the result be denoted $BetP^{\Theta}\{S_v^X\}$.

5. Compute $Info_X(S)$, as Quinlan:

$$Info_X(S) = \sum_{v \in D(X)} \frac{|S_v^X|}{|S|} Info(S_v^X) \tag{18}$$

where $D(X)$ is the domain of the possible values of the attribute X and $Info(S_v^X)$ is computed using $BetP^{\Theta}\{S_v^X\}$.
6. Compute the information gain provided by the attribute X in the set of objects S such that:

$$Gain(S,X) = Info(S) - Info_X(S) \tag{19}$$

7. Using the split Info, compute the gain ratio relative to attribute X:

$$Gain\ Ratio(S,X) = \frac{Gain(S,X)}{SplitInfo(S,X)} \tag{20}$$

where

$$Split\ Info(S,X) = -\sum_{v \in D(X)} \frac{|S_v^X|}{|S|} \log_2 \frac{|S_v^X|}{|S|} \tag{21}$$

8. Repeat the same process for every attribute X belonging to the set of attributes that can be selected. Next, choose the one that maximizes the gain ratio.

4.1.1.2. Conjunctive approach. The conjunctive approach is based on an intra-group distance quantifying for each attribute value how strongly objects are close from each others. The different steps upon this attribute selection measure ensuring the building of a decision tree are the following ones:

1. For each training object, compute:

$$K^{\Theta}\{I_j\}(C) = -\ln q^{\Theta}\{I_j\}(C) \quad \forall C \subseteq \Theta \tag{22}$$

from the bba $m^{\Theta}\{I_j\}$.
2. For each attribute value v of an attribute X, compute the joint $K^{\Theta}\{S_v^X\}$ defined on Θ, the set of possible classes by

$$K^{\Theta}\{S_v^X\} = \sum_{I_j \in S_v^X} K^{\Theta}\{I_j\} \tag{23}$$

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

$$SumD(S_v^X) = \frac{1}{|S_v^X|} \sum_{I_j \in S_v^X} \sum_{C \subseteq \Theta} (K^{\Theta}\{I_i\}(C) - \frac{1}{|S_v^X|} K^{\Theta}\{S_v^X\}(C))^2 \tag{24}$$

Thanks to the property of the commonality functions, the conjunctive combination rule can be written as a product, and even better, the logarithm of commonality functions of combination is the sum of the logarithms of the commonality functions entered into the conjunctive combination.
4. Compute $SumD_X(S)$ representing the weighted sum of the different $SumD(S_v^X)$ relative to each value v of the attribute X:

$$SumD_X(S) = \sum_{v \in D(X)} \frac{|S_v^X|}{|S|} SumD(S_v^X) \tag{25}$$

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

$$Diff(S,X) = SumD(S) - SumD_X(S) \tag{26}$$

where

$$SumD(S) = \frac{1}{|S|}\sum_{I_j\in S}\sum_{C\subseteq\Theta}\left(K^{\Theta}\{I_i\}(C) - \frac{1}{|S|}K^{\Theta}\{S\}(C)\right)^2 \tag{27}$$

6. Using the Split Info, compute the diff ratio relative to the attribute X:

$$Diff\ \ Ratio(S,X) = \frac{Diff(S,X)}{Split\ Info(S,X)} \tag{28}$$

7. For every attribute repeat the same process, and choose the one that maximizes the diff ratio.

4.1.2. Partitioning strategy

The partitioning strategy for the construction of a belief decision tree is similar to the partitioning strategy used in the classical tree. Since we deal with only symbolic attributes, we create an edge for each value of the attribute chosen as a decision node.

4.1.3. Stopping criteria

Four strategies are proposed as stopping criteria:

1. If the treated node includes only one instance.
2. If the treated node includes only instances for which the $m^{\Theta}\{I_j\}$'s are equal.
3. If all the attributes are split.
4. If the value of the applied attribute selection measure (using either the gain ratio or the diff ratio) for the remaining attributes is less or equal than zero.

4.1.4. Structure of leaves

Each leaf in the induced tree will be characterized by a bba. According to the used attribute selection measure:

- Using the averaging approach, the leaf's bba is equal to the average of the bba's of the objects belonging to this leaf.
- Using the conjunctive approach, the leaf's bba is the result of combination of the bba's of objects belonging to this leaf using the conjunctive rule.

4.2. Belief decision tree procedures

Like the standard decision tree, the belief decision tree is composed of two principal procedures: the building or the construction of the tree from training objects with uncertain classes and the classification of new instances that may be characterized by uncertain or even missing attribute values.

4.2.1. Building procedure

Building a decision tree in this context of uncertainty will follow the same steps presented in C4.5 algorithm. Furthermore, this algorithm is generic since it offers two possibilities for selecting attributes by using either the averaging approach or the conjunctive one.

4.2.2. Classification procedure

Once the belief decision tree is constructed, it is able to classify an object described by an exact value for each one of its attributes [7], we have to start from the root of the belief decision tree, and repeat to test the attribute at each node by taking into account the attribute value until reaching a leaf. As a leaf is characterized by a bba on classes, the pignistic transformation is applied to get the pignistic probability on the classes of the object to classify in order to decide its class. For instance, one can choose the class having the highest pignistic probability. Belief decision trees also deal with the classification of new instances characterized by uncertainty in the values of their attributes. The idea to classify such objects is to look for the leaves that the given instance may belong to by tracing out possible paths induced by the different attribute values of the object to classify. The new instance may belong to many leaves where each one is characterized by a basic belief assignment. These bba's are combined using disjunctive rule of combination in order to get beliefs on the instance's classes.

5. Pruning belief decision tree methods

A belief decision tree is a classification technique based on decision trees within the framework of belief function theory. Inducing a belief decision tree may lead in most cases to very large trees with bad classification accuracy and difficult comprehension. Several pruning methods have been developed to cope with this problem including minimal cost-complexity pruning [3], reduced error pruning [18], critical value pruning [15], pessimistic error pruning [18], minimum error pruning [16] and error based pruning [21]. All these methods deal with only standard decision trees and not with BDT. So, our objective is to adapt one of these post-pruning methods in order to simplify the belief decision tree and improve its classification accuracy. In our work, we will choose minimal cost-complexity pruning (MCCP) to adapt for pruning belief decision trees. This pruning method is appealing because it performs well in terms of size pruned tree and accuracy. It also produces a selection of trees for the expert to study. It is helpful if several trees, pruned to different degrees are available.

This section is dedicated to the presentation of our pruning belief decision tree methods in averaging and conjunctive approaches based on MCCP. We start by explaining how this method works in a certain case, then we present our pruning methods in an uncertain case and in both approaches.

5.1. Minimal cost-complexity pruning in certain case

The MCCP, was developed by Breiman et al. [3]. This method is also known as the CART pruning algorithm. It consists of two steps:

1. Generating a series of increasingly pruned trees $\{T_0, T_1, T_2, \ldots, T_n\}$.
2. Selecting the best tree with the lowest error rate on separate test set.

As regarding the first step, T_{i+1} is obtained from T_i by pruning all the nodes having the lowest increase in error rate per pruned leaf denoted α.

$$\alpha = \frac{R(t) - R(T_t)}{NT - 1} \tag{29}$$

where NT is the number of leaves in a node and $NT - 1$ is the number of pruned leaves.

- $R(t)$ is the error rate if the node is pruned, which becomes a leaf belonging to only one class. It is the proportion of training examples which do not belong to this class.
- $R(T_t)$ is the error rate if the node is not pruned. It represents the average of the error rates at the leaves weighted by the number of examples at each leaf.

The method works as follows:

1. Compute α for each (non-terminal) node (except the root) in T_i.
2. Prune all the nodes with the smallest value of α, so obtaining the tree T_{i+1}.
3. Repeat this process until only root is left yields a series of pruned tree.
4. The next step is to select one of these as the final tree. The criterion for selection of the final tree is the lowest mis-classification rate on independent data set. This selection is based only on testing set accuracy.

5.2. Minimal cost-complexity pruning in uncertain case

In this section we will propose our pruning belief decision tree methods in averaging and conjunctive approaches based on standard minimal cost-complexity pruning (MCCP).

5.2.1. MCCP in averaging approach

Our objective is to develop a pruning method based on standard minimal cost-complexity pruning to prune belief decision tree in averaging approach. To prune a node in MCCP, we compute the error rate if the node is pruned or not. To do this, we must know at each node or leaf, the number of objects belonging to each class. However, in a belief decision tree, the class of the objects are represented by a basic belief assignment and not by a certain class.

The idea is to use the pignistic transformation. It is a function which can transform the belief function to probability function in order to make decisions from beliefs. This function is used to build the belief decision tree in averaging approach. So, we will transform beliefs on classes to a distribution of probability at each node or leaf to know the proportion of each class.

In this section, we propose the following steps to prune the belief decision tree by adapting MCCP.

1. For each node in the belief decision tree, compute the pignistic probability of each object I_j by applying the pignistic transformation to $m^{\Theta}\{I_j\}$.
2. Compute the sum pignistic probability function $BetP^{\Theta}\{S\}$ taken over the set of objects S belonging to a node N. For each $C_i \in \Theta$,

$$BetP^{\Theta}\{S\}(C_i) = \sum_{I_j \in S} BetP^{\Theta}\{I_j\}(C_i)$$

$BetP^{\Theta}\{S\}$ represents the number of objects belonging to each class $C_i \in \Theta$ for a node N. In this way, we can compute the number of errors of each node. It is the sum of objects not allocated to the class which occurs most frequently.

3. Compute the error rate if the node is pruned, which becomes a leaf.

$$R(t) = \frac{\sum_{C_i \in \Theta}(BetP^{\Theta}\{S\}(C_i)) - \mathrm{Max}(BetP^{\Theta}\{S\}(C_i))}{|T|} \tag{30}$$

where the sum of $BetP^{\Theta}\{S\}(C_i)$ represents the number of objects belonging to the node. $\mathrm{Max}(BetP^{\Theta}\{S\}(C_i))$ represents the number of training objects belonging to the class which occurs most frequently. So, the difference is the number of errors and $|T|$ is the number of training objects.

4. Compute the error rate if the node is not pruned.

$$R(T_t) = \sum R(i) \quad \text{for } i = \text{sub-treeleaves} \tag{31}$$

5. Compute the increase in error per pruned leaf, denoted α_ave.

$$\alpha\_\mathrm{ave} = \frac{R(t) - R(T_t)}{NT - 1} \tag{32}$$

where NT represents the number of leaves in the node and $NT - 1$ is the number of pruned leaves.

6. Repeat the same process for every node in the belief decision tree only the root.

If a node has the lowest α_ave, starting pruned it and obtaining the first pruned tree. The node becomes a leaf represented by the average bba relative to the objects belong to it.

Continue this process until the root is left, yields a series of pruned trees.

In the selection phase, from the series of pruned trees select the best tree with the lowest error rate on testing set.

Example 1. To explain how to compute the value of α_ave of a node in an uncertain context, we take a node N containing three instances and has two leaves. I_1 and I_2 belong to leaf $F1$ and I_3 belongs to the other leaf F2. This node is taken from a BDT induced from training set of 10 instances. The class of each object is represented by a bba $m^{\Theta}\{I_j\}$ are defined as follows:

$$m^{\Theta}\{I_1\}(C_1) = 0.7; \quad m^{\Theta}\{I_1\}(\Theta) = 0.3;$$
$$m^{\Theta}\{I_2\}(C_1) = 0.6; \quad m^{\Theta}\{I_2\}(\Theta) = 0.4;$$
$$m^{\Theta}\{I_3\}(C_1) = 0.95; \quad m^{\Theta}\{I_3\}(\Theta) = 0.05.$$

Compute the pignistic probability of each object I_j (see Table 1).

The node N has 2.5 objects of class C_1, and 0.25 of class C_2 and 0.25 of C_3. The must frequently class is C1 and 0.5 is the number of errors. So, with pignistic transformation we can make a decision.

The error rate if the node is pruned (see Eq. (30)):

$$R(t) = \frac{3 - 2.5}{10} = 0.05;$$

Table 1
Computation of $BetP^{\Theta}\{S\}$

	C_1	C_2	C_3
$BetP^{\Theta}\{I_1\}$	0.8	0.1	0.1
$BetP^{\Theta}\{I_2\}$	0.74	0.13	0.13
$BetP^{\Theta}\{I_3\}$	0.96	0.02	0.02
$Sum = BetP^{\Theta}\{S\}$	2.5	0.25	0.25

The error rate if the node is not pruned (see Eq. (31)):

$$R(T_t) = R(F1) + R(F2) = \frac{0.46}{10} + \frac{0.04}{10} = 0.05;$$

The increase in error per pruned leaf, denoted α_ave (see Eq. (32)):

$$\alpha\_ave = 0.$$

5.2.2. MCCP in conjunctive approach

In this section, we propose our second pruning belief decision tree method based on the conjunctive approach. This method should be closer to the TBM itself like in the building method. The conjunctive approach of building BDT is based on an intra-group distance used in the attribute selection measure and quantifying for each attribute how strongly objects are close from each others. Knowing what 'nice' property should be satisfied by the instances in a leaf once the tree is build. Ideally, they should belong to the same class, but their actual classes are unkonwn. So, all objects in a leaf have bba's that are close to each others. Thus, a distance between bba's is required.

With MCCP in a certain case, we start pruning the nodes having the lowest increase in error rate on training set if it will be pruning (nodes have the minimum α) because pruning a node leads to an increase in error rate on training set. So by analogy, our idea is to start pruning nodes having the lowest increase of distance between bba objects on classes with pruning. Since, pruning will lead to an increase in the distance between bba's objects on classes. The distance between bba's objects in a node is superior than the distance of its leaves.

We propose the following steps to prune belief decision tree in conjunctive approach based on MCCP.

1. Compute the distance if the node is pruned, denoted by $D(t)$. It represents the sum of distances separating each bba training instance on classes belonging to the node. This distance between training objects is used in the attribute selection measure for the conjunctive approach [9].

$$D(t) = \frac{1}{|S|} \sum_{I_j \in S} \sum_{C \subseteq \Theta} \left(K^{\Theta}\{I_j\}(C) - \frac{1}{|S|} K^{\Theta}\{S\}(C) \right)^2 \tag{33}$$

where

$$K^{\Theta}\{I_j\}(C) = -\ln q^{\Theta}\{I_j\}(C) \quad \forall C \subseteq \Theta \tag{34}$$

2. Compute the distance if the node is not pruned, denoted by $D(T_t)$: the sum of distance between objects in the leaves. Pruning leads to an increase in the distance between bba's objects.
3. Divide this increase by the number of leaves, we obtain the increase in distance by pruned leaf, denoted by α_conj.

$$\alpha\_\text{conj} = \frac{D(t) - D(T_t)}{NT - 1} \tag{35}$$

where NT is the number of leaves in the node and $NT - 1$ is the number pruned leaves.
4. Repeat the same process for every node in the belief decision tree only the root.If a node has the lowest α_conj starting pruned it and obtaining the first pruned tree. The node becomes a leaf represented by the conjunctive bba relative to the objects belong to it.

Continue this process until the root is left, yields a series of pruned trees. In the selection phase, from the series of pruned trees select the best tree with the lowest error rate on testing set.

Example 2. Let us continue with the last example to explain how compute the value of α_conj for the same node N composed of three objects I_1, I_2 and I_3 and has two leaves. The class of each object is represented by a bba $m^\Theta\{I_j\}$.

The distance between bba's objects if the node is pruned:

$$D(t) = \frac{1}{3}\sum_{I_j \in S}\sum_{C \subseteq \Theta}\left(K^\Theta\{I_j\}(C) - \frac{1}{3}K^\Theta\{S\}(C)\right)^2 = 5.078;$$

The distance between bba's objects if the node is not pruned:

$$D(T_t) = D(F1) + D(F2) = 0.1241 + 0 = 0.1241;$$

where $D(F1)$ is the distance between bba's of the instances I_1 and I_2 belonging to the leaf $F1$, and $D(F2)$ is equal to 0 because the leaf $F2$ has one instance.

The increase in distance per pruned leaf:

$$\alpha\_conj = \frac{5.078 - 0.1241}{2 - 1} = 4.9541;$$

6. Experimentation and simulation

In our experiments, we have performed several tests and simulations on real databases obtained from the U.C.I. repository and have only symbolic attributes: Wisconsin Breast Cancer database, Balance Scale Weight, Congressional Voting, Solar, Zoo and Nursery databases available in.[1] These databases have different numbers of cases from 101 instances to 12960 instances, different numbers of attributes from 3 to 17 attributes and different numbers of classes from two classes to seven classes.

These databases are modified in order to include uncertainty in classes.

[1] http://www.ics.uci.edu/mlearn/MLRepository.html.

6.1. Constructing uncertainty in databases

Instances with partially known classes are usually eliminated from the databases. The belief decision trees are essentially built to handle uncertain classes where their uncertainty is represented by a bba given on the set of possible classes. These bba's are created artificially. They take into account three basic parameters:

- The real class C of an object.
- Degree of uncertainty
  - No uncertainty: we take $P = 0$
  - Low degree of uncertainty: we take $0 \prec P \preceq 0.3$
  - Middle degree of uncertainty: we take $0.3 \prec P \preceq 0.6$
  - High degree of uncertainty: we take $0.6 \prec P \preceq 1$

For each object's class, its bba has almost two focal elements:

1. The first is the actual class C of the object with bbm, $m(C) = 1 - P$ (P is a probability generated randomly).
2. The second is a subset θ of Θ (generated randomly) such that the actual class of the object under consideration belongs to θ and every of the other class belongs to θ with probability P. $m(\theta) = P$.

A larger P gives a larger degree of uncertainty.

6.2. Evaluation criteria

The relevant criteria used to judge the performance of our pruning methods are as follows:

1. *The size* characterized by the number of nodes and leaves in the belief decision tree.
2. *The PCC* represents the percent of correct classification of the objects belonging to testing set. In the framework of belief decision trees, leaves are charecterized by bba's. For each testing instance, we determine its corresponding leaf and we look for the most probable class corresponding to this leaf using the pignistic probability computed from the leaf's bba.
3. *The distance criterion* allowing to take into account all the beliefs characterizing the leaf's bba. It compares the pignistic probability induced from the testing instance's bba and its real class [11].

The distance criterion for a testing instance I_j belonging to a leaf L (its bba is $m^\Theta\{L\}$) is defined as follows:

$$dist\_crit(I_j) = Distance(BetP^\Theta\{L\}, C(I_j)) = \sum_{i=1}^{n}(BetP^\Theta\{L\}(C_i) - \delta_{j,i})^2 \tag{36}$$

where the real class of the testing instance I_j is $C(I_j)$, and $\delta_{j,i} = 1$ if $C(I_j) = C_i$ and 0 otherwise.

This distance verifies the following property:

$$0 \leqslant dist\_crit(I_j) \leqslant 2 \tag{37}$$

Next, we have to compute the average total distance relative to all the classified testing instances denoted dist_crit. So, we get:

$$dist = \frac{\sum_{I_j \text{ in classified instances}} dist(I_j)}{\text{total number of classified instances}} \tag{38}$$

The PCC and distance are considered as classification accuracy criteria. Accurate BDT with high PCC and low distance.

6.3. Results

Different results carried out from these simulations will be presented and analyzed in order to evaluate our proposed pruning methods for certain and uncertain cases.

1. *The certain case:* The first case tests the efficiency of the pruning method in Quinlan algorithm when there is no uncertainty in classes by applying the averaging approach.
2. *The uncertain case:* The second case tests the efficiency of our pruning methods in both averaging and conjunctive approaches with many levels of uncertainty (low, middle and high).

Let us remind that our objective is to reduce the size and improve the classification accuracy of belief decision tree based on averaging and conjunctive approaches by pruning it. So, we compare the size, PCC and distance of BDT without pruning, with pre-pruning and with applying our post-pruning methods. In the introduction, we mentioned that there are two pre-pruning procedures applied on BDT. The first presented in [10] and the second in [6]. We will choose for comparison the first procedure to be coherent with averaging and conjunctive approaches of building.

Each data set is divided into 10 parts. Nine parts are used as the training set, the last is used as the testing set. The procedure is repeated 10 times, each time another part is chosen as the testing set. This method called a cross-validation permits an unbiased estimation of the evaluation criterion. So, the mean of size, PCC and distance of BDT of 10 times K-10 folds cross-validation is presented in experimental results.

6.3.1. Results of the certain case

Tables 2–4 summarize the different results relative to Cancer, Balance, Vote, Solar, Zoo and Nursery databases for the certain case. So, we compare the mean of size, PCC and the distance criterion of BDT without pruning (Size.bef.Prun, PCC.bef.Prun, Dist.bef.Prun), with pre-pruning (Size.aft.Pr.Prun, PCC.aft.Pr.Prun, Dist.aft.Pr.Prun) and with applying our post-pruning method in averaging approach (Size.aft.Pt.Prun, PCC.aft.Pt.Prun, Dist.aft.Pt.Prun).

From these tables, we can conclude that our pruning method in certain case has good results for each evaluation criteria. From Table 2, there are an improvement of the mean size of the tree in all databases. For example, in Cancer database, the mean size goes from 274 items to 123 items. For Balance database, the mean size of the induced tree goes from

Table 2
Experimental measures [certain case (mean size)]

Database	Size		
	bef.Prun	aft.Pr.Prun	aft.Pt.Prun
Cancer	274	151	123
Balance	326	265	109
Vote	57	34	29
Solar	421	301	256
Zoo	89	53	47
Nursery	398	261	176

Table 3
Experimental measures [certain case (mean PCC)]

Database	PCC		
	bef.Prun (%)	aft.Pr.Prun (%)	aft.Pt.Prun (%)
Cancer	76.6	77.2	82.53
Balance	60.7	62.3	70.9
Vote	95.21	95.74	96.53
Solar	80	82.64	86.13
Zoo	85.21	86.74	90.08
Nursery	93.66	94.14	96.07

Table 4
Experimental measures [certain case (mean distance)]

Database	Dist		
	bef.Prun	aft.Pr.Prun	aft.Pt.Prun
Cancer	0.35	0.33	0.29
Balance	0.44	0.43	0.43
Vote	0.24	0.22	0.21
Solar	0.32	0.29	0.27
Zoo	0.31	0.29	0.26
Nursery	0.26	0.25	0.23

326 items to 109 items. For Vote database, the mean size is reduced from 57 items to 29 items.

From Table 3, there are also an increase of the mean PCC for all databases. For example, in Balance database, the mean PCC is increased from 60.7% to 70.9%. For Solar, the mean PCC goes from 80% to 86.13%. For Zoo database, the mean PCC is improved from 85.21% to 90.08%.

From Table 4, there are an improvement of distance criterion for all databases. For Cancer database, there is a good decrease of mean distance from 0.35 to 0.29. For Nursery database, the distance is improved from 0.26 to 0.23. For Balance database, the mean distance is sightly decreased from 0.44 to 0.43.

From these tables, we can also conclude that pre-pruning reduces size, increases PCC and decreases distance criterion for all databases, but not better than our post-pruning method. For example in Cancer database, the size goes from 274 items to 151 items, the PCC is increased from 76.6% to 77.2% and the distance is improved from 0.35 to 0.33.

6.3.2. Results of the uncertain case

This section presents different results carried out from testing our pruning belief decision tree methods in averaging and conjunctive approaches on uncertain case.

6.3.2.1. Size. Tables 5–10 compare the mean size of BDT before pruning (Size.bef.Prun_ave, Size.bef.Prun_conj) with the mean size after pre-pruning (Size.aft.Pr.Prun_ave, Size.aft.Pr.Prun_conj) and the mean size after our post-pruning methods (Size.aft.Pt.Prun_ave, Size.aft.Pt.Prun_conj) in both approaches.

Table 5
Experimental measures [cancer, uncertain case (mean size)]

Degree of uncertainty	Size.bef		Size.aft			
	Prun_ave	Prun_conj	Pr.Prun_ave	Pr.Prun_conj	Pt.Prun_ave	Pt.Prun_conj
Low degree	399	352	302	292	82	81
Middle degree	401	357	305	285	87	90
High degree	444	425	321	265	101	125
Mean	414	378	309	280	90	98

Table 6
Experimental measures [balance, uncertain case (mean size)]

Degree of uncertainty	Size.bef		Size.aft			
	Prun_ave	Prun_conj	Pr.Prun_ave	Pr.Prun_conj	Pt.Prun_ave	Pt.Prun_conj
Low degree	508	489	338	323	139	125
Middle degree	498	456	331	308	147	138
High degree	522	502	373	325	97	109
Mean	509	482	347	318	127	124

Table 7
Experimental measures [vote, uncertain case (mean size)]

Degree of uncertainty	Size.bef		Size.aft			
	Prun_ave	Prun_conj	Pr.Prun_ave	Pr.Prun_conj	Pt.Prun_ave	Pt.Prun_conj
Low degree	171	202	133	142	75	70
Middle degree	158	173	124	131	56	58
High degree	157	171	111	125	58	67
Mean	162	182	122	132	63	65

Table 8
Experimental measures [solar, uncertain case (mean size)]

Degree of uncertainty	Size.bef		Size.aft			
	Prun_ave	Prun_conj	Pr.Prun_ave	Pr.Prun_conj	Pt.Prun_ave	Pt.Prun_conj
Low degree	465	483	306	312	215	227
Middle degree	472	496	298	301	224	236
High degree	489	498	317	316	237	245
Mean	475	492	306	309	225	236

Table 9
Experimental measures [zoo, uncertain case (mean size)]

Degree of uncertainty	Size.bef		Size.aft			
	Prun_ave	Prun_conj	Pr.Prun_ave	Pr.Prun_conj	Pt.Prun_ave	Pt.Prun_conj
Low degree	95	97	63	69	57	61
Middle degree	98	102	71	73	58	57
High degree	113	106	76	75	58	59
Mean	102	101	70	72	57	59

Table 10
Experimental measures [nursery, uncertain case (mean size)]

Degree of uncertainty	Size.bef		Size.aft			
	Prun_ave	Prun_conj	Pr.Prun_ave	Pr.Prun_conj	Pt.Prun_ave	Pt.Prun_conj
Low degree	456	463	224	230	178	192
Middle degree	453	461	206	213	171	184
High degree	471	469	237	235	166	187
Mean	460	464	222	226	171	187

From these tables, our pruning belief decision tree methods in both approaches work well for all degrees of uncertainty for all databases. They lead to a reduction in size better than pre-pruning.

From Table 5, the mean size of BDT in averaging approach for Cancer database goes from 414 items to 309 items with pre-pruning and to 90 items with our pruning method (Fig. 1). In conjunctive approach, the mean size goes from 378 items to 280 items with pre-pruning and to 90 items with our pruning method (Fig. 2).

For Balance database, is the same thing. The mean size is reduced from 509 items to 347 with pre-pruning and reduced to 127 items with our pruning method in averaging approach. In conjunctive approach, The size is reduced from 482 items to 318 with pre-pruning and reduced to 124 items with our pruning method in averaging approach.

For Vote database, the Table 7 shows that our pruning belief decision tree methods have good results on belief decision tree. The size is improved from 162 items to 122 items

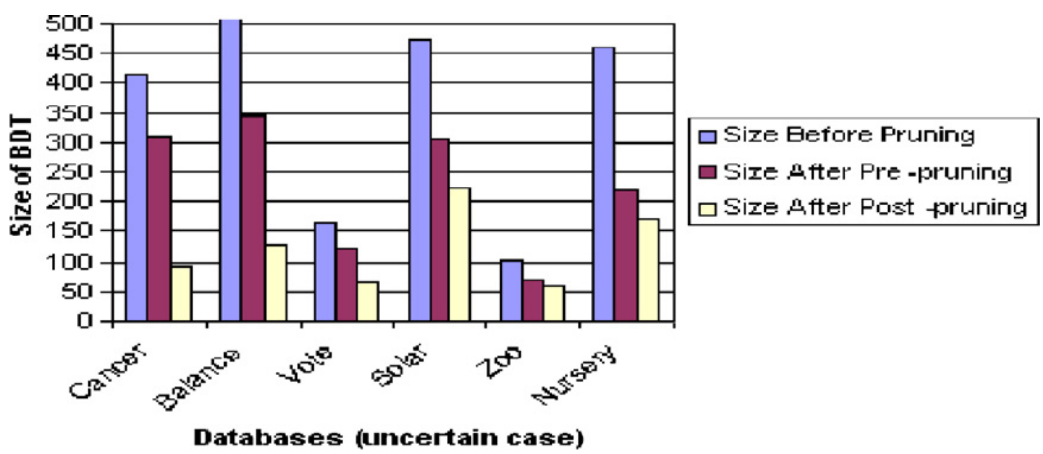

Fig. 1. The mean size of BDT in averaging approach.

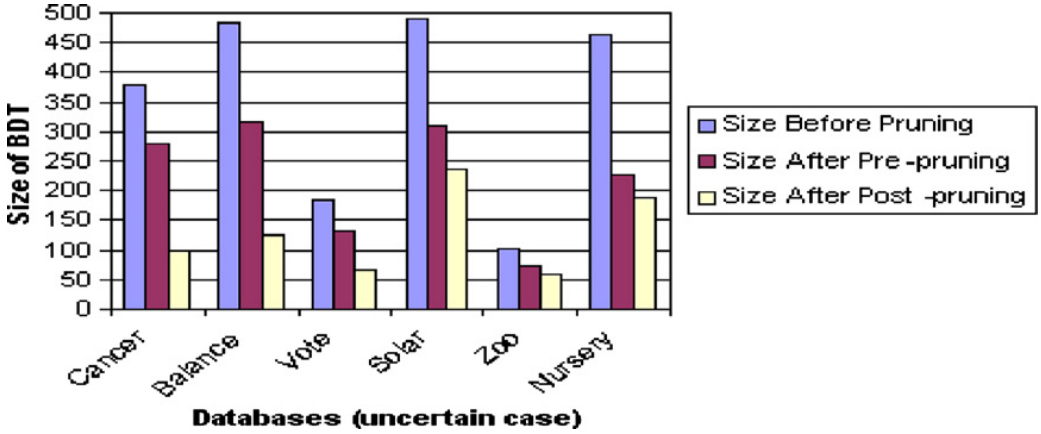

Fig. 2. The mean size of BDT in conjunctive approach.

with pre-pruning and improved to 63 items in averaging approach. The size is improved from 182 items to 132 items with pre-pruning and improved to 65 items in conjunctive approach.

From these tables, we can also conclude that the size after our post-pruning methods in both approaches is almost the same for all databases. For example, in Vote database, there are 63 items in averaging approach and 65 items in conjunctive approach.

6.3.2.2. PCC. Tables 11–16 compare the mean PCC of BDT before pruning (PCC.bef.Prun_ave, PCC.bef.Prun_conj) with the mean PCC after pre-pruning (PCC.aft.Pr.Prun_ave, PCC.aft.Pr.Prun_conj) and the mean PCC after our post-pruning methods (PCC.aft.Pt.Prun_ave, PCC.aft.Pt.Prun_conj) in both approaches.

Table 11
Experimental measures [cancer, uncertain case (mean PCC)]

Degree of uncertainty	PCC.bef		PCC.aft			
	Prun_ave (%)	Prun_conj (%)	Pr.Prun_ave (%)	Pr.Prun_conj (%)	Pt.Prun_ave (%)	Pt.Prun_conj (%)
Low degree	67.97	68.81	69.13	71.13	82.38	83.46
Middle degree	67.83	68.18	68.38	70.76	82.54	83.01
Migh degree	66.09	68.1	67.18	70.03	81.29	82.17
Mean	67.29	68.36	68.23	70.64	82.07	82.88

Table 12
Experimental measures [balance, uncertain case (mean PCC)]

Degree of uncertainty	PCC.bef		PCC.aft			
	Prun_ave (%)	Prun_conj (%)	Pr.Prun_ave (%)	Pr.Prun_conj (%)	Pt.Prun_ave (%)	Pt.Prun_conj (%)
Low degree	58.68	63.41	66.84	68.01	77.81	78.15
Middle degree	58.35	63.36	66.58	67.26	77.78	77.83
High degree	63.1	63.23	66.26	65.84	81.7	83.76
Mean	60	63.33	66.56	67.03	79	79.9

Table 13
Experimental measures [vote, uncertain case (mean PCC)]

Degree of uncertainty	PCC.bef		PCC.aft			
	Prun_ave (%)	Prun_conj (%)	Pr.Prun_ave (%)	Pr.Prun_conj (%)	Pt.Prun_ave (%)	Pt.Prun_conj (%)
Low degree	94.29	95	94.88	95.78	96.78	98.28
Middle degree	94.08	94.88	94.29	95.21	96	97.76
High degree	92.27	92.47	92.65	93.02	95.33	97.71
Mean	93.67	94.11	93.94	94.43	96	97.91

Table 14
Experimental measures [solar, uncertain case (mean PCC)]

Degree of uncertainty	PCC.bef		PCC.aft			
	Prun_ave (%)	Prun_conj (%)	Pr.Prun_ave (%)	Pr.Prun_conj (%)	Pt.Prun_ave (%)	Pt.Prun_conj (%)
Low degree	78.2	78.51	81.68	81.73	85.28	85.72
Middle degree	77.9	78.23	81.39	81.64	84.97	85.61
High degree	77.87	77.91	81.05	81.02	84.66	85.46
Mean	77.99	78.21	81.36	81.46	84.97	85.59

Table 15
Experimental measures [zoo, uncertain case (mean PCC)]

Degree of uncertainty	PCC.bef		PCC.aft			
	Prun_ave (%)	Prun_conj (%)	Pr.Prun_ave (%)	Pr.Prun_conj (%)	Pt.Prun_ave (%)	Pt.Prun_conj (%)
Low degree	85.09	85.12	86.78	86.81	90.65	91.94
Middle degree	84.42	84.61	85.92	86.23	89.74	91.36
High degree	83.26	83.43	86.05	86.11	90.02	91.41
Mean	84.25	84.38	86.25	86.38	90.13	91.57

Table 16
Experimental measures [nursery, uncertain case (mean PCC)]

Degree of uncertainty	PCC.bef		PCC.aft			
	Prun_ave (%)	Prun_conj (%)	Pr.Prun_ave (%)	Pr.Prun_conj (%)	Pt.Prun_ave (%)	Pt.Prun_conj (%)
Low degree	93.69	93.72	94.58	94.73	95.68	95.84
Middle degree	93.01	93.18	94.19	94.26	94.92	95.13
High degree	92.27	92.29	93.65	93.71	94.73	95.11
Mean	92.99	93.06	94.14	94.23	95.11	95.36

From these tables, we can conclude that our pruning belief decision tree methods in both approaches lead to an increase PCC for all degrees of uncertainty and for all databases better than pre-pruning.

For Balance database, the mean PCC from Table 12 is increased from 60% to 66.56% with pre-pruning and increased to 79% with our pruning method in averaging approach.

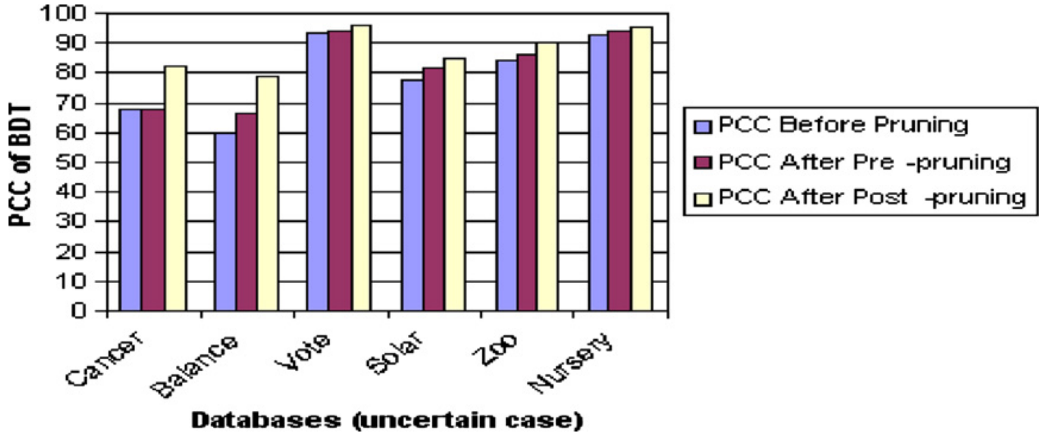

Fig. 3. The mean PCC of BDT in averaging approach.

In conjunctive approach, the mean PCC is increased from 63.33% to 67.03% with pre-pruning and increased to 77.9% with our pruning method in averaging approach.

For Solar database, the Table 14 shows that our pruning belief decision tree methods have good results on BDT. The PCC is improved from 77.99% to 81.36% with pre-pruning and improved to 84.97% with our pruning method in averaging approach (Fig. 3). The PCC goes from 78.21% to 81.46% with pre-pruning and goes to 85.59% with our pruning method in conjunctive approach (Fig. 4).

For Zoo database, the mean PCC is increased from 84.25% to 86.25% with pre-pruning and increased to 90.13% with our pruning method in averaging approach. In conjunctive approach, the mean PCC is increased from 83.38% to 86.38% with pre-pruning and increased to 91.57% with our pruning method in averaging approach.

From these tables, we can conclude that the mean PCC in conjunctive approach is sightly better than in averaging approach for all databases. For example, in Vote database, the mean PCC before pruning is 93.67% in averaging approach and 94.11% in conjunctive approach. After pre-pruning, the PCC in averaging approach is 93.94% and 94.43% in the second approach. After our pruning method, the PCC is 96% in averaging approach and 97.91% in the other approach. Besides, we can also conclude that in most cases an increase in the degree of uncertainty leads to a decrease in the PCC.

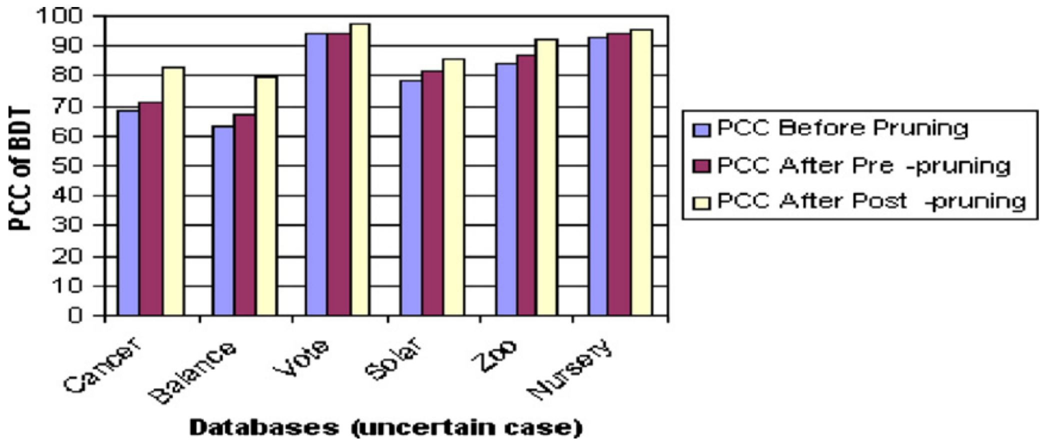

Fig. 4. The mean PCC of BDT in conjunctive approach.

6.3.2.3. Distance criterion: dist. Tables 17–22 compare the mean distance of BDT before pruning (Dist.bef.Prun_ave, Dist.bef.Prun_conj) with the mean distance after pre-pruning (Dist.aft.Pr.Prun_ave, Dist.aft.Pr.Prun_conj) and the mean distance after our post-pruning methods (Dist.aft.Pt.Prun_ave, Dist.aft.Pt.Prun_conj) in both approaches.

From these tables, our pruning belief decision tree methods in both approaches work well for all degrees of uncertainty and for all databases. They lead to a reduction in distance better than pre-pruning.

Table 17
Experimental measures [cancer, uncertain case (mean distance)]

Degree of uncertainty	Dist.bef		Dist.aft			
	Prun_ave	Prun_conj	Pr.Prun_ave	Pr.Prun_conj	Pt.Prun_ave	Pt.Prun_conj
Low degree	0.38	0.36	0.36	0.34	0.29	0.27
Middle degree	0.39	0.39	0.36	0.37	0.32	0.31
High degree	0.41	0.4	0.37	0.37	0.34	0.32
Mean	0.39	0.38	0.36	0.36	0.31	0.3

Table 18
Experimental measures [balance, uncertain case (mean distance)]

Degree of uncertainty	Dist.bef		Dist.aft			
	Prun_ave	Prun_conj	Pr.Prun_ave	Pr.Prun_conj	Pt.Prun_ave	Pt.Prun_conj
Low degree	0.39	0.38	0.38	0.37	0.35	0.34
Middle degree	0.45	0.42	0.43	0.41	0.41	0.38
High degree	0.46	0.45	0.44	0.43	0.41	0.4
Mean	0.43	0.41	0.41	0.4	0.39	0.37

Table 19
Experimental measures [vote, uncertain case (mean distance)]

Degree of uncertainty	Dist.bef		Dist.aft			
	Prun_ave	Prun_conj	Pr.Prun_ave	Pr.Prun_conj	Pt.Prun_ave	Pt.Prun_conj
Low degree	0.24	0.23	0.23	0.21	0.2	0.19
Middle degree	0.26	0.24	0.25	0.23	0.21	0.21
High degree	0.27	0.25	0.25	0.23	0.22	0.21
Mean	0.25	0.24	0.24	0.22	0.21	0.2

Table 20
Experimental measures [solar, uncertain case (mean distance)]

Degree of uncertainty	Dist.bef		Dist.aft			
	Prun_ave	Prun_conj	Pr.Prun_ave	Pr.Prun_conj	Pt.Prun_ave	Pt.Prun_conj
Low degree	0.33	0.32	0.29	0.28	0.26	0.25
Middle degree	0.34	0.33	0.31	0.3	0.27	0.25
High degree	0.34	0.34	0.3	0.3	0.27	0.26
Mean	0.33	0.33	0.3	0.29	0.26	0.25

Table 21
Experimental measures [zoo, uncertain case (mean distance)]

Degree of uncertainty	Dist.bef		Dist.aft			
	Prun_ave	Prun_conj	Pr.Prun_ave	Pr.Prun_conj	Pt.Prun_ave	Pt.Prun_conj
Low degree	0.32	0.31	0.3	0.3	0.24	0.25
Middle degree	0.32	0.3	0.29	0.28	0.25	0.25
High degree	0.33	0.32	0.31	0.3	0.26	0.24
Mean	0.32	0.31	0.3	0.29	0.25	0.24

Table 22
Experimental measures [nursery, uncertain case (mean distance)]

Degree of uncertainty	Dist.bef		Dist.aft			
	Prun_ave	Prun_conj	Pr.Prun_ave	Pr.Prun_conj	Pt.Prun_ave	Pt.Prun_conj
Low degree	0.27	0.27	0.25	0.22	0.2	0.2
Middle degree	0.28	0.27	0.25	0.24	0.21	0.21
High degree	0.28	0.26	0.25	0.24	0.22	0.21
Mean	0.27	0.25	0.25	0.23	0.21	0.2

In averaging approach, the mean distance of BDT from Cancer database goes from 0.39 to 0.36 with pre-pruning and to 0.31 with our pruning method. In conjunctive approach, the mean distance goes from 0.38 to 0.36 with pre-pruning and to 0.3 with our pruning method. For Balance database, is the same thing. The distance is reduced from 0.43 to 0.41 with pre-pruning and to 0.39 with our pruning method in averaging approach. In conjunctive approach, The distance is improved from 0.41 to 0.4 with pre-pruning and to 0.37 with our pruning method.

For Vote database, the Table 19 shows that our pruning belief decision tree methods have good results on belief decision tree. The distance decreased from 0.25 to 0.24 with pre-pruning and to 0.21 with our pruning method in averaging approach (Fig. 5). The distance is improved from 0.24 to 0.22 with pre-pruning and to 0.2 with our pruning method in conjunctive approach (Fig. 6).

From these tables, we can also conclude that the distance in conjunctive approach is sightly better than the distance in averaging approach for all databases. For example, in

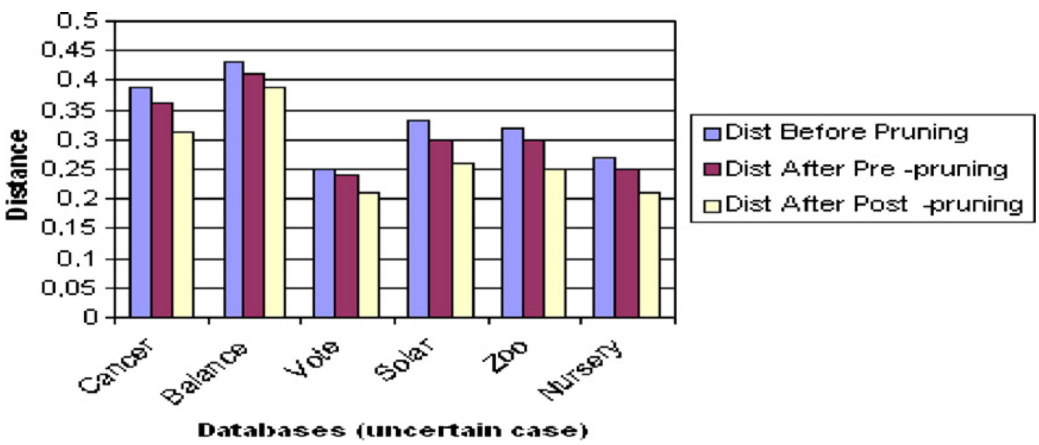

Fig. 5. The mean distance of BDT in averaging approach.

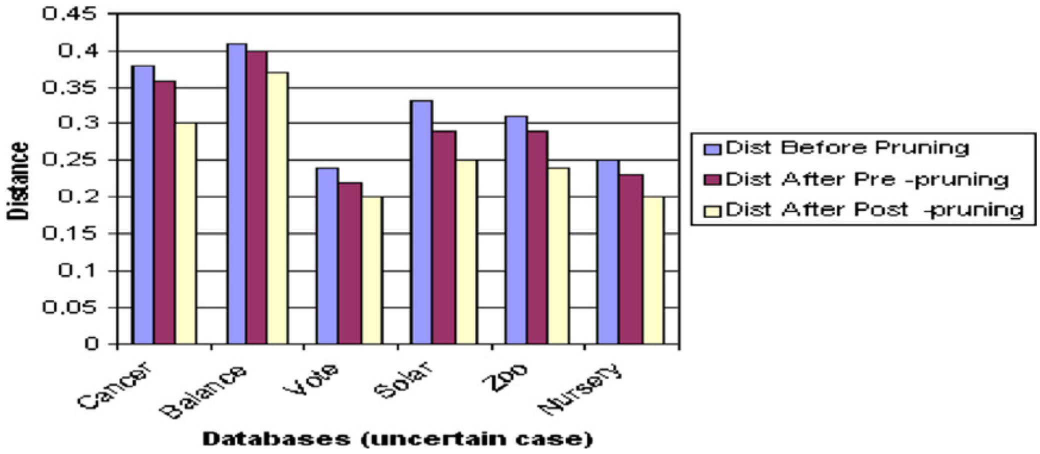

Fig. 6. The mean distance of BDT in conjunctive approach.

Balance database, the distance before pruning is 0.43 in averaging approach and 0.41 in conjunctive approach. After pre-pruning is 0.41 in averaging approach and 0.4 in second approach. After post-pruning is 0.39 in averaging approach and is 0.37 in the other approach.

6.3.3. Computational complexity

In this section, we describe the complexity analysis of the three procedures to construct BDT. The complexity of building the tree without pruning and with pre-pruning is $O(mn \log n)$ where n is the number of training instances and m is the number of attributes. With post-pruning, we add the complexity of MCCP $O(N)$ which N is the number of nodes in the tree. The table 23 shows the time complexity of learning BDT without pruning, with pre-pruning and after post-pruning for the different databases. Note that the time complexity is almost the same for the different degrees of uncertainty and with very small differences between averaging and conjunctive approaches.

From this table, we can conclude that the time complexity of the construction of BDT with pre-pruning is the lowest. It is better than without pruning and with post-pruning for all databases, this is explained by the fact that the pre-pruning approach controls the growing of the BDT. On the other hand, the time complexity of the construction of BDT with post-pruning is obviously the highest since, post-pruning approach starts the pruning of the tree after the building. However, the difference between the time complexity of the three procedures is not very high for all databases. The differences are 20 s, 16 s, 6 s,

Table 23
Experimental measures (time complexity of learning BDT)

Database	Time (s)		
	bef.Prun	aft.Pr.Prun	aft.Pt.Prun
Cancer	136	120	156
Balance	123	112	139
Vote	111	105	117
Solar	142	131	160
Zoo	91	80	103
Nursery	323	245	386

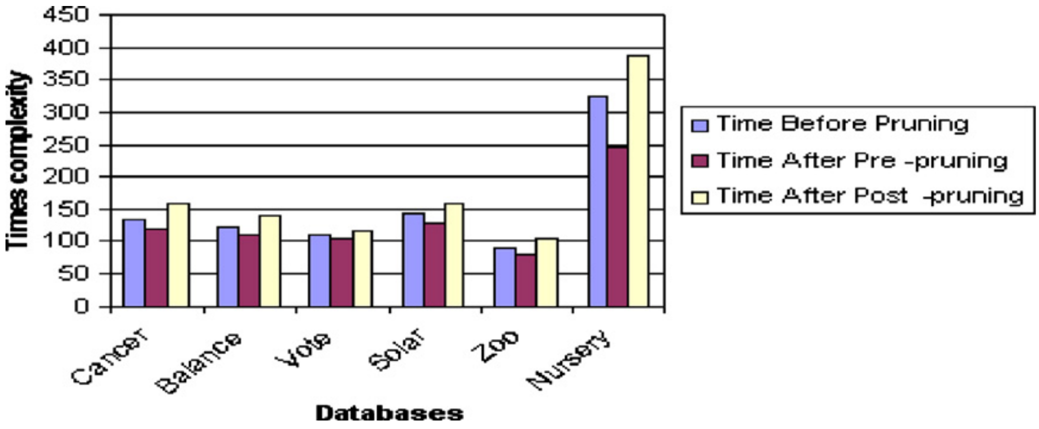

Fig. 7. The time complexity of learning BDT.

18 s, 12s and 63 s respectively for Cancer, Balance, Vote, Solar, Zoo and Nursery databases between building BDT after post-pruning and without pruning (Fig. 7).

Note that for the classification phase, we have found that the time complexity using BDT after post-pruning is the best since the post-pruning gives a best reduction in size more than pre-pruning.

7. Conclusion

In this paper, we have presented our pruning belief decision tree methods in averaging and conjunctive approaches with the objective to reduce the size of the induced tree and improve the classification accuracy in an uncertain context. Pruning is a way to cope with the problem of overfitting. Then, we have presented the different results obtained from simulations and that have been performed on real databases. These experimentations have shown interesting results for the performance of our post-pruning methods comparing with BDT without pruning and with pre-pruning.

Regarding the interesting results obtained in this work, we could propose further works that may be done to improve our pruning method. So, we propose to develop other pruning BDT methods based on other standard pruning methods. we can also apply the post-pruning approach to the BDT developed in [6,30,31] and compare it with their pre-pruning strategy. It will be also interesting to extend the belief decision tree approach to handle continuous attributes and the uncertainty in attributes values in the training set.

Acknowledgements

The authors would like to thank Thierry Denoeux for his interest in this work and for his help. The authors dedicate this paper to the memory of the late Patrick Vannoorenberghe, for the interesting discussions they had with him during the writing of this paper. Many thanks are also due to anonymous referees for their detailed comments, criticisms, and suggestions that have helped greatly in improving the quality of this paper.

References

[1] J.A. Barnett, Calculating Dempster–Shafer plausibility, IEEE Transactions on Pattern Analysis and Machine Intelligence 13 (6) (1991) 599–602.

[2] N. Ben Amor, S. Benferhat, Z. Elouedi, Qualitative classification with possibilistic decision trees, in: Proceedings of the International Conference on Information Processing of Uncertainty in Knowledge Based Systems IPMU'2004, vol. 2, Perugia, Italia, 2004, pp. 881–887.
[3] L. Breiman, J.H. Friedman, R.A. Olshen, C.J. Stone, Classification and Regression Trees, Wadsworth, Belmont, CA, 1984.
[4] A.P. Dempster, A generalisation of Bayesian inference, Journal of the Royal Statistical Society, Series B 30 (1968) 205–247.
[5] T. Denoeux, Reasoning with imprecise belief structures, International Journal Approximate Reasoning, IJAR 20 (1999) 70–111.
[6] T. Denoeux, M. Skarstien-Bajanger, Induction of decision trees form partially classified data using belief functions, in: Proceedings of the IEEE International Conference on Systems, vol. 4, Nashiville, USA, 2000, pp. 2923–2928.
[7] Z. Elouedi, K. Mellouli, P. Smets, Decision trees using the belief function theory, in: Proceedings of the International Conference on Information Processing and Management of Uncertainty IPMU'2000, vol. 1, Madrid, Spain, 2000, pp. 141–148.
[8] Z. Elouedi, K. Mellouli, Induction of belief decision trees: a conjunctive approach, in: G. Govaert, J. Janssen, N. Limnios (Eds.), Proceedings of the Tenth Conference of the Applied Stochastic Models and Data Analysis, ASMDA, Compiégne, France, 2001, pp. 404–409.
[9] Z. Elouedi, K. Mellouli, P. Smets, Belief decision trees: theoretical foundations, International Journal of Approximate Reasoning, IJAR 28 (2001) 91–124.
[10] Z. Elouedi, K. Mellouli, P. Smets, A pre-pruning method in belief decision trees, in: The Ninth International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems IPMU 2002, vol. 1, Annecy, France, 2002, pp. 579–586.
[11] Z. Elouedi, P. Vannorenberghe, Inférence statistique et modèle de croyances transférables, les Rencontres francophones sur la Logique Floue et ses Applications conference LFA 2006, Toulouse, France, 2006. pp. 61–68.
[12] U.M. Fayyad, K.B. Irani, On the handling of continuous-valued attributes in decision tree generation, Machine Learning 8 (1992) 87–102.
[13] I. Jenhani, Z. Elouedi, N. Ben Amor, K. Mellouli, Qualitative inference in possibilistic option decision trees, in: Proceedings of the Eight European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty ECSQARU-2005, Barcelone, Spain, 2005, pp. 944–955.
[14] E. Hüllermeier, Possibilistic induction in decision tree learning, in: Proceedings of the 13th European Conference on Machine Learning ECML2002, Helsinki, Finland, 2002, pp. 173–184.
[15] J. Mingers, Expert systems – rule induction with statistical data, Journal of the Operational Research Society 8 (1987) 39–47.
[16] T. Niblett, I. Bratko, Learning decision rules in noisy domains, Proceedings of Expert Systems, vol. 86, Cambridge University Press, Cambridge, 1986.
[17] J.R. Quinlan, Induction of decision trees, Machine Learning 1 (1986) 81–106.
[18] J.R. Quinlan, Simplifying decision trees, International Journal of Man–Machine Studies (1987) 221–234.
[19] J.R. Quinlan, Decision trees as probabilistic classifiers, in: Proceedings of the Fourth International on Machine Learning, Morgan Kaufman, 1987, pp. 31–37.
[20] J.R Quinlan, Probabilistic decision tree, in: R.S. Michalski, J.G. Carbonell, T.M Michell (Eds.), Machine Learning, vol. 3, 1990, pp. 267–301 (Chapter 5).
[21] J.R. Quinlan, C4.5: Programs for Machine learning, Morgan Kaufman, San Mateo, CA, 1993.
[22] G. Shafer, A Mathematical Theory of Evidence, Princeton University Press, Princeton, NJ, 1976.
[23] C.E. Shannon, A mathematical theory of communication, The Bell Systems Technical Journal 27 (3) (1948) 379–423, 623–656.
[24] P. Smets, Belief functions, in: P. Smets, E.H. Mamdani, D. Dubois, H. Prade (Eds.), Non Standard Logics for Automated Reasoning, Academic Press, London, 1988, pp. 253–286.
[25] P. Smets, R. Kennes, The transferable belief model, Artificial Intelligence 66 (1994) 191–236.
[26] P. Smets, The application of belief transferable belief model to diagnostic problems, International Journal of Intelligent Systems 13 (1998) 127–157.
[27] P. Smets, The transferable belief model for quantified belief representation, in: D.M. Gabbay, P. Smets (Eds.), Handbook of Defeasible Reasoning and Uncertainty Management Systems, vol. 1, Kluwer, Doordrecht, The Netherlands, 1998, pp. 207–301.

[28] M. Umano, H. Okamoto, I.H. Tamura, F. Kawachi, S. Umededzu, J. Kinoshita, Fuzzy decision trees by fuzzy ID3 algorithm and its application to diagnosis systems, in: Proceedings of the Third IEEE International Conference on Fuzzy Systems, vol. 3, FUZZ- IEEE'94, Orlondo, USA, 1994, pp. 2113–2118.

[29] P.E. Utgoff, ID5: an incremental ID3, in: Proceedings of the Fifth International Conference on Machine Learning, Morgan Kaufman, 1988, pp. 107–120.

[30] P. Vannoorenberghe, T. Denoeux, Handling uncertain labels in multiclass problems using belief decision trees, in: IPMU'2002, 2002, pp. 1916–1926.

[31] P. Vannoorenberghe, On aggregating belief decision trees, Information Fusion 5 (2) (2004) 179–188.

[32] J. Zeidler, M. Schlosser, Continuous valued attributes in fuzzy decision trees, in: Proceedings of the Sixth International Conference on Information Processing and Management of Uncertainty, vol. 1, Granada, Spain, 1996, pp. 395–400.