New decision tree classifier for dealing with partially uncertain data

Nouvelle méthode d'arbre de décision pour le traitement des données partiellement incertaines

A. Trabelsi[1,2] Z. Elouedi[1] E. Lefevre[2]

[1] Université de Tunis, Institut Supérieur de Gestion de Tunis, LARODEC , Tunisia

[2] Univ. Artois, EA 3926, Laboratoire de Génie Informatique et d'Automatique de l'Artois (LGI2A), 62400 Béthune, France

trabelsyasma@gmail.com
zied.elouedi@gmx.fr
eric.lefevre@univ-artois.fr

Résumé :

L'arbre de décision est l'une des méthodes de classification les plus connues et a été largement utilisé dans plusieurs domaines, notamment dans la fouille de données et l'apprentissage automatique. Cependant, les algorithmes standards de construction de l'arbre de décision ne sont pas capables de gérer l'incertitude, en particulier l'incertitude épistémique. Dans cet article, nous proposons d'adapter la technique d'arbre de décision à un environnement incertain. Concrètement, nous traitons le cas où les valeurs d'attributs d'objets sont incertaines et où cette incertitude est représentée dans le cadre de la théorie des fonctions de croyance. La méthode proposée concerne à la fois la construction de l'arbre de décision et la classification de nouveaux objets.

Mots-clés :

Arbres de décision, théorie des fonctions de croyance, classification, incertitude.

Abstract:

Decision tree is one of the most common classification approaches and has been widely used in several fields, notably in Data Mining and Machine Learning. However, the classical decision tree building algorithms have not the ability to handle uncertainty, in particular the epistemic one. In this paper, we propose to adapt the decision tree technique to an uncertain environment. Concretely, we tackle the case where object's attribute values are uncertain and where this uncertainty is represented within the belief function framework. Our proposed method will concern both the decision tree building task and the classification of new objects.

Keywords:

Decision trees, belief function theory, classification, uncertainty.

1 Introduction

Decision trees are regarded as ones of the efficient machine learning techniques that are widely used in the artificial intelligence field. Their success is notably explained by their ability to provide understandable representations easily interpreted by experts and even by ordinary users (non-specialist readers). Numerous decision tree building approaches have been developed [1, 7]. Such algorithms take as input a training set that includes objects characterized by a set of known attribute values as well as their assigned classes and outputs a decision tree enabling the classification of new objects. In several real world applications, data may be uncertain due to some factors such as data randomness, data incompleteness, etc. However, classical decision tree versions are not able to handle uncertainty that may occur in the attribute values or in the class values for a given classification problem. This shortcoming has led to the introduction of the fuzzy decision trees [15], the possibilistic decision trees [4], the uncertain decision trees [6] and the probabilistic decision trees [8]. Although the probability theory is widely used for modeling uncertainty, several researchers have proved that probability cannot always be the adequate tool for representing data uncertainty, concretely epistemic uncertainty. The belief function theory, also called the Dempster-Shafer theory, has the advantage to represent all kinds of knowledge availability [2]. The process of incorporating belief function theory within the decision tree learning algorithm has been already developed

[3, 14, 16]. In the best of our knowledge, all existing decision tree versions under the belief function framework handle only the case of uncertain class labels. However, in several real-world cases, uncertainty may also occur in the attribute values [10]. Inspired from the belief decision tree paradigm proposed in [3] as an extension of the standard decision tree of Quinlan [8] to tackle data with uncertain class labels, in this paper, we develop a novel decision tree classifier within the Transferable Belief Model (TBM) [12] to handle uncertainty that may exist in the attribute values. Precisely, we tackle both the construction and the classification phases. The reminder of this paper is organized as follows: Section 2 is devoted to highlighting the fundamental concepts of the belief function theory as interpreted by the TBM framework. In Section 3, we detail our decision tree building procedure from partially uncertain data. Section 4 presents our classification scenario. Our experimentation on several real world databases are given in Section 5. Finally, in Section 6, we draw our conclusion and our main future work directions.

2 Belief function theory

In this Section, we briefly recall the fundamental concept underlying the TBM framework [12], one interpretation of the belief function theory.

2.1 Frame of discernment

Let Θ be a finite non-empty set of N elementary events related to a given problem. Such set Θ, called the frame of discernment, is set as:

$$\Theta = \{\theta_1, \theta_2, \ldots, \theta_N\} \tag{1}$$

The power set of Θ, denoted by 2^Θ, is composed of all the subsets of Θ. It is defined as follows:

$$\begin{aligned} 2^\Theta &= \{Y, Y \subseteq \Theta\} \\ &= \{\emptyset, \theta_1, \theta_2, ..., \theta_N, \{\theta_1, \theta_2\}, ..., \Theta\} \end{aligned}$$

2.2 Basic belief assignment

The beliefs held by a such agent over the subsets of the frame of discernment Θ are represented by the so called basic belief assignment (bba), denoted by m, such that:

$$\begin{gathered} m : 2^\Theta \to [0, 1] \\ \sum_{Y \subseteq \Theta} m(Y) = 1 \end{gathered} \tag{2}$$

The quantity $m(Y)$, called basic belief mass (bbm), states the degree of belief committed exactly to the event Y. All subsets $Y \subseteq \Theta$ such that $m(Y) > 0$ are called focal elements.

2.3 Combination rules

Suppose that m_1 and m_2 are two basic belief assignments provided by fully reliable distinct information sources [11] and defined in the same frame of discernment Θ. The combination of m_1 and m_2 through the conjunctive rule is set as:

$$(m_1 \textcircled{\cap} m_2)(X) = \sum_{Y, Z \subseteq \Theta : Y \cap Z = X} m_1(Y).m_2(Z) \tag{3}$$

It is substantial to note that some cases require the combination of bbas defined on different frames of discernment. Let Θ_1 and Θ_2 be two frames of discernment, the idea is to extend Θ_1 and Θ_2 to a joint frame of discernment Θ. This process, known under the name of vacuous extension, is defined as:

$$\Theta = \Theta_1 \times \Theta_2 \tag{4}$$

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:

$$m^{\Theta_1 \uparrow \Theta}(X) = m_1(Y) \; where \; X = Y \times \Theta_2, \; Y \subseteq \Theta_1 \tag{5}$$

$$m^{\Theta_1 \uparrow \Theta}(X) = 0 \; otherwise$$

2.4 Distance between two pieces of evidence

The distance measures represent the degree of dissimilarity between bodies of evidence. In recent years, many researchers have investigated a set of different approaches to compute distance between a set of bbas [5, 9, 13]. The Jousselme distance, proposed in [5], is regarded as one of the most commonly used distances. Let m_1 and m_2 be two pieces of evidence, the Jousselme distance between m_1 and m_2 is computed as follows:

$$dist(m_1, m_2) = \sqrt{\frac{1}{2}(m_1 - m_2)^T D(m_1 - m_2)} \tag{6}$$

where D is the Jaccard similarity measure defined by:

$$D(X, Y) = \begin{cases} 1 & \text{if } X{=}Y{=}\emptyset \\ \dfrac{|X \cap Y|}{|X \cup Y|} & \forall\, X{,}Y \in 2^{\Theta} \end{cases} \tag{7}$$

3 Learning Decision tree classifier from partially uncertain data

Since real world applications may incorporate uncertainty, the process of constructing machine learning algorithms from uncertain data has attracted the attention of several researchers over the past few years. Authors, in [3], have suggested the so-called belief decision tree as an extension of the standard decision tree for dealing with databases described by uncertain class labels, precisely where the uncertainty is represented within the belief function framework. However, in several domains, the uncertainty may also appear in the attribute values. Thus, it is important to develop a classification model to handle databases characterized by uncertain attribute values. We detail, in this Section, our proposed decision tree classifier for treating the problem of uncertain attribute values within the TBM framework where we have used the following notations:

— T: a given training set composed by M objects O_i; $i = \{1,\ldots,M\}$.

— S: a subset of objects belonging to the training set T.

— $C= \{C_1,\ldots, C_Q\}$: represents the Q possible classes of the classification problem.

— $A{=}\{A_1,\ldots,A_n\}$: the set of n attributes.

— Θ^{A_k}: corresponds to all the possible values of an attribute $A_k \in A$ where $k = \{1, \ldots, n\}$.

— ${m_i}^{\Theta^{A_k}}$: is the bba relative to the attribute A_k of the object O_i.

— ${m_i}^{\Theta^{A_k}}(v)$: denotes the bbm assigned to the hypothesis that the actual attribute value of object O_i belongs to $v \subseteq \Theta^{A_k}$.

— $L= \{L_1,\ldots, L_F\}$: represents the F generated leaves when building the decision tree.

3.1 Decision Tree parameters for partially uncertain data

We highlight, in this Section, the main parameters enabling the construction of our proposed decision tree within the belief function framework. Firstly, we present our proposed attribute selection measure. Then, we detail the splitting strategy process and we point out the different stopping criteria. Finally, we describe the structure of leaves in an uncertain context.

Attribute selection measure. The attribute selection measure is regarded 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 best separate the training instance into homogeneous subsets. In this paper, we are based on an 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 attribute:

1. We compute the total distance taken over the training set T as follows:

$$SumD(S) = \sum_{O_i \in S} \sum_{O_{j \geq i+1} \in S} \frac{1}{|n|} \sum_{k=1}^{n} P_i^S . P_j^S dist(m_i^{\Theta_{A_k}}, m_j^{\Theta_{A_k}}) \tag{8}$$

where $dist$ represents the Jousselme distance between two bbas and P_i^S states the probability of belonging of the object O_i to the set S. It is calculated as the cross product of the pignistic probabilities of the different attribute'values bbas relative to the object O_i and allowing O_i to belong to S.

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

$$SumD(S_v^{A_k}) = \sum_{O_i \in S_v^{A_k}} \sum_{O_{j \geq i+1} \in S_v^{A_k}} P_i^{S_v^{A_k}} .P_j^{S_v^{A_k}} dist(m_i^{\Theta_{A_k}}, m_j^{\Theta_{A_k}}) \quad (9)$$

where $P_i^{S_v^{A_k}}$ quantifies the probability of the object O_i to belong to the subset $S_v^{A_k}$. Note that it is computed as the same manner as the computation of P_j^s.

3. Once the different $SumD(S_v^{A_k})$ are calculated, for each attribute $A_k \in A$, we compute $SumD_{A_k}(S)$ as follows:

$$SumD_{A_k}(S) = \sum_{v \in \Theta^{A_k}} SumD(S_v^{A_k}) \quad (10)$$

4. In analogy to classical decision trees, we compute the difference before and after the partitioning process has been performed using the attribute A_k. This measure, denoted by $diff(S, A_k)$, is defined as the difference between $SumD(S)$ and $SumD_{A_k}(S)$ as follows:

$$diff(S, A_k) = SumD(S) - SumD_{A_k}(S) \quad (11)$$

5. Using the $SplitInfo$, we compute the $DiffRatio$ relative to the attribute A_k.

$$DiffRatio(S, A_k) = \frac{diff(S, A_k)}{SplitInfo(S, A_k)} \quad (12)$$

where

$$SplitInfo(S, A_k) = \quad (13)$$
$$- \sum_{v \in D(A_k)} \frac{|S_v^{A_k}|}{|S|} log_2 \frac{|S_v^{A_k}|}{|S|}$$

6. Repeat this process for each attribute A_k $\in A$ and then select the one that maximize the $DiffRatio$.

Splitting Strategy. The splitting strategy consists of dividing the training set according to the values of the attribute chosen as a decision node. We associate an edge for each attribute value. Consequently, we obtain several training subsets where each subset includes objects having the same attribute value. As we deal with uncertain attribute values, each training object may be part of more than one subset with a probability of belonging computed in term of the pignistic probability of the object's attribute bbas.

Stopping criteria. The stopping criteria are quite 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 gain ratio equal or less than zero.

Structure of leaves. Our ultimate purpose is to construct decision tree from data characterized by uncertain attribute values. In such case an object O_i may belong to more than one leaf with a probability of belonging denoted by P_i^{Lf}. As leaves may include objects with different class values, our proposed decision tree building algorithm assigns for each leaf a probability distribution over the set of classes computed from the probability of objects belonging to this leaf. The probability distribution relative to L_f over a class $C_q \in C$ is set as:

$$Pr\{L_f\}(C_q) = \frac{1}{\sum_{O_i \in L_f} P_i^{L_f}} \sum_{O_i \in L_f} P_i^{L_f} \gamma_{iq} \quad (14)$$

where γ_{iq} equals 1 if the class of the object O_i is C_q, 0 otherwise and $P_i^{L_f}$ is the probability of the instance O_i to belong to the leaf L_f. This latter is calculated as the cross product of the pignistic probabilities of the object O_i to belong to the nodes that link the root node and the corresponding leaf node L_f.

3.2 The construction of a decision tree classifier from partially uncertain table

The construction of our proposed decision tree classifier within an uncertain environment will follow the same Quinlan's algorithm steps which requires a top down approach for constructing standard decision tree. Assume that T is our learning set, the different steps of our decision tree learning algorithm will be as follows:

1. We start by creating the root node from the whole learning 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.
   - else, we pick out the attribute that maximizes the attribute selection measure presented previously. The chosen one will be the root node of our decision tree relative to the set T.
3. We create a branch for each attribute value chosen as a root. This partitioning step leads to several subsets where each one contains as much as possible homogenous objects according to the attribute value.
4. We restart the same process from level 2 until all nodes are considered as leaves.

It is worth noting that the complexity of building our proposed decision tree is $O(n{*}m^2{*}log\ m)$ where m is the number of training instances and n is the number of attributes.

4 Classification scenario within an uncertain context

As stated by Quinlan [8], a decision tree paradigm consists mainly in two distinct procedures which are the construction and the classification steps. In this Section, we propose a novel approach for classifying objects with uncertain attributes. Let J be the total number of testing instances O_j $(j = \{1, \dots, J\})$ and $A = \{A_1, \dots, A_n\}$ be the set of n attributes describing 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} as follows:

$$\Theta^A = \underset{k=1,\dots,n}{\times} \Theta^{A_k}. \tag{15}$$

Since objects are described by a combination of values where each of them corresponds to one attribute, we have firstly to compute for each object to be classified the joint bba expressing beliefs on its attribute values. To perform 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 operator:

$$m_j^{\Theta^A} = \underset{k=1,\dots,n}{\bigcirc\!\!\!\!\!\cap}\ m_j^{\Theta^{A_k}\uparrow\Theta^A} \tag{16}$$

If our joint bba $m_j^{\Theta^A}$ is obtained, we move on to compute the probability distribution $Pr_j[x](C_q)$ of each focal element x of an object O_j $(q = \{1, \dots, Q\})$. It will be noted that the computation of this probability distribution depends mainly on the focal elements of the bba m^{Θ^A} and on the subset x. This dependency is expressed in what follows:

- When x is a singleton, the probability distribution $Pr_j[x](C_q)$ will be equal to the probability of the leaf's class C_q for which the focal element is attached.
- else if the focal element is not a singleton, we explore all possible paths cor-

respond to this combination of values. There are two possible cases:

  - The case 1 is that all paths lead to the same leaf. In this case the probability $Pr_j[x](C_q)$ will be equal to the probability of the leaf's class C_q for which the focal element is attached.
  - The case 2 is that paths lead to distinct leaves. In this case the probability $Pr_j[x](C_q)$ will be equal to the average probability of the class C_q relative to the different attached leaves.

- Finally, the probability distribution relative to each object test O_j over the set of classes will be set as:

$$Pr_j(C_q) = \sum_{x \subseteq \Theta^A} m^{\Theta^A}(x) Pr_j[x](C_q)$$
$$\forall\, q \in \{1, \ldots, Q\} \text{ and } j = \{1, \ldots, J\} \tag{17}$$

The most probable class of the object O_j is the one having the highest probability $Pr_j(C)$.

5 Implementation and simulation

This Section is devoted to detailing our experimentation tests that we have carried out to assess the performance of our proposed decision tree classifier.

5.1 Experimentation settings

Regarding time complexity of our proposed approach, we have performed our experiments on real small categorical databases obtained from the UCI repository. A brief description of these databases is presented in Table 1.

Since, in this paper, we only deal with uncertain attributes to construct our decision tree classifier, we propose to generate synthetic databases from the original ones by including uncertainty. We have tackled different uncertainty levels according to a degree of uncertainty denoted by P:

- Certain Case: P=0
- Low Uncertainty: $0.1 \leq P < 0.4$
- Middle Uncertainty: $0.4 \leq P < 0.7$
- High Uncertainty: $0.7 \leq P \leq 1$

As previously mentioned, each attribute value v of an object O_i such that $v \subseteq \Theta^{A_k}$ should be expressed by a bba as follows:

$$\begin{aligned} m_i^{\Theta^{A_k}}(v) &= 1 - P \\ m_i^{\Theta^{A_k}}(\Theta^{A_k}) &= P \end{aligned} \tag{18}$$

To check the performance of our proposed decision tree method when classifying new objects, we have relied on two evaluation criteria, namely the Percentage of Correctly Classification criterion (PCC) and a distance criterion. The former one quantifies the percent of correct classification of objects belonging to the test set. It is set as follows:

$$PCC = \frac{\text{Number of well classified instances}}{\text{Number of classified instances}} \tag{19}$$

where the $Number\ of\ well\ classified\ instances$ represents the number of test instances for which the most probable classes yielded through our novel decision tree learning algorithm are the same as the real ones and the $Number\ of\ classified\ instances$ corresponds to the total number of classified instances.

The latter one is used in the purpose of performing a comparison between each test instance's probability distribution over the set of classes and its real class. It is computed as follows where γ_{jq} equals 1 when C_q represents the real class of the test instance O_j and 0 otherwise:

$$DistanceCriterion_j = \sum_{q=1}^{Q} (Pr_j(C_q) - \gamma_{jq})^2 \tag{20}$$

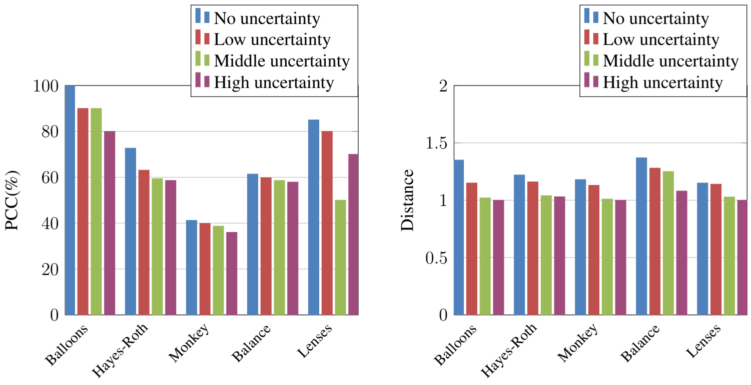
Note that this distance satisfies the following property:

$$0 \leq DistanceCriterion_j \leq 2 \tag{21}$$

Besides, we just have to compute the average distance yielded from all test instances to get a total distance.

Tableau 1 – Description of databases

Databases	#Instances	#Attributes	#Classes
Balloons	16	4	2
Hayes-Roth	160	5	3
Monkey	432	7	2
Balance	625	4	3
Lenses	24	4	3



(a) PCC results

(b) Distance results

Figure 1 – Experimentation results

5.2 Experimentation results

We have performed 10-fold cross-validation approach to ensure an efficient estimation of the evaluation criteria. This paradigm splits each database into ten parts: nine parts are used as the training set and the remaining part is used as the testing set. This procedure will be repeated ten times where each part is used exactly once as a test set. It is substantial to note that our approach for building decision trees does not give the same result as the Quinlan C4.5 algorithm in the case of no-uncertainty in the attribute values. This is explained by the fact that the C4.5 algorithm uses the $GainRatio$ as a splitting criterion, while our proposed classifier relies on the $DiffRatio$ criterion. These two mentioned splitting criteria may generate different decision tree structures. So, as a first experimentation, we start by comparing our proposed approach with Quinlan's algorithm in term of the PCC criterion. The classification results are given in Table 2 where we can remark that our proposed approach has given the best classification accuracy for all the mentioned databases compared to the C4.5 algorithm. Accordingly, we can deduce that the splitting criterion may affect the accuracy of the classifier. Then, we have applied our proposed decision tree in several uncertainty cases. Our experimental results in terms of the PCC and the distance criteria are depicted respectively in Figure 1(a) and Figure 1(b) for the different mentioned databases. From Figure 1(a), we can notice that our proposed decision tree algorithm has yielded good classification results in term of the PCC criterion for the different uncertainty levels of the different databases. For example, for Hayes-Roth database, we have 72.7%, 63.1%, 59.4% and 58.6% as PCCs relative respectively to no, low, middle and high uncertainties. Regarding the distance criterion, from Figure 1(b), we remark that our proposed classifier has also

Tableau 2 – Comparing the Quinlan and the co-uncertainty cases according to the PCC criterion

Databases	No-uncertainy Case (%)	Quinlan Case (%)
Balloons	100	80
Hayes-Roth	72.7	57
Monkey	41.2	39.07
Balance	61.4	50
Lenses	85	70

given interesting results. In fact, all distance values belong to the closed interval [1,1.37]. For example, the distance results relative to Lenses database are 1.15, 1.14, 1.08 and 1 for respectively no, low, middle and high uncertainties. We can also remark that both the PCC and the distance values decrease with the increasing of the uncertainty degree in the most of cases.

6 Conclusion

In this paper, we have developed a novel decision tree algorithm for tackling epistimic uncertainty that occurs in the attribute values. Although we have yielded interesting results in terms of the PCC and the distance criteria, time complexity is stilling a challenging problem for large or even medium sized databases. Thus, as a future work, we intend to minimize time complexity by reducing the dimensionality space and also by applying a pruning procedure. Also, we intend to develop other extensions of our proposed approach to handle the case of numerical and mixed databases.

Références

[1] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth and Brooks, Monterey, CA, 1984.

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

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

[4] E. Hüllermeier. Possibilistic induction in decision-tree learning. In *Machine Learning: ECML 2002*, pages 173–184. Springer, 2002.

[5] A. Jousselme, D. Grenier, and E. Bossé. A new distance between two bodies of evidence. *Information fusion*, 2(2):91–101, 2001.

[6] B. Qin, Y. Xia, and F. Li. Dtu: a decision tree for uncertain data. In *Advances in Knowledge Discovery and Data Mining*, pages 4–15. Springer, 2009.

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

[8] J. R. Quinlan. Decision trees as probabilistic classifiers. In *4th international machine learning*, pages 31–37, 1987.

[9] B. Ristic and P. Smets. The TBM Global Distance Measure for the Association of Uncertain Combat ID Declarations. *Information Fusion*, 7(3):276–284, 2006.

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

[11] P. Smets. Application of the transferable belief model to diagnostic problems. *International journal of intelligent systems*, 13(2-3):127–157, 1998.

[12] P. Smets and R. Kennes. The transferable belief model. *Artificial intelligence*, 66(2):191–234, 1994.

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

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

[15] M. Umano, H. Okamoto, I. Hatono, H. Tamura, F. Kawachi, S. Umedzu, and J. Kinoshita. Fuzzy decision trees by fuzzy id3 algorithm and its application to diagnosis systems. In *3rd IEEE Conference on Fuzzy Systems*, pages 2113–2118. IEEE, 1994.

[16] P. Vannoorenberghe and T. Denoeux. Handling uncertain labels in multiclass problems using belief decision trees. In *Proceedings of IPMU*, volume 3, pages 1919–1926, 2002.