Handling Uncertain Attribute Values In Decision Tree Classifier Using The Belief Function Theory

Asma Trabelsi[1], Zied Elouedi[1], and Eric Lefevre[2]

[1] Université de Tunis, Institut Supérieur de Gestion de Tunis, LARODEC , Tunisia
trabelsyasma@gmail.com,zied.elouedi@gmx.fr
[2] Univ. Artois, EA 3926, Laboratoire de Génie Informatique et d'Automatique de l'Artois (LGI2A), Béthune, F-62400, France
eric.lefevre@univ-artois.fr

Abstract. Decision trees are regarded as convenient machine learning techniques for solving complex classification problems. However, the major shortcoming of the standard decision tree algorithms is their unability to deal with uncertain environment. In view of this, belief decision trees have been introduced to cope with the case of uncertainty present in class' value and represented within the belief function framework. Since in various real data applications, uncertainty may also appear in attribute values, we propose to develop in this paper another version of decision trees in a belief function context to handle the case of uncertainty present only in attribute values for both construction and classification phases.

Keywords: Decision trees - uncertain attribute values - belief function theory - classification

1 Introduction

Decision trees are one of the well known supervised learning techniques applied in a variety of fields, particulary in artificial intelligence. Indeed, decision trees have the ability to deal with complex classification problems by producing understandable representations easily interpreted not only by experts but also by ordinary users and providing logical classification rules for the inference task. Numerous decision tree building algorithms have been introduced over the years [**?**, **?**, **?**]. Such algorithms take as inputs a training set composed with objects described by a set of attribute values as well as their assigned classes and output a decision tree that enables the classification of new objects. A significant shortcoming of the classical decision trees is their inability to handle data within an environment characterized by uncertain or incomplete data. In the case of missing values, several kinds of solutions are usually considered. One of the most popular solutions is dataset preprocessing strategy which aims at removing the missing values. Other solutions are exploited by some systems implementing decision tree learning algorithms. Missing values may also be considered as a particular case of uncertainty and can be modeled by several uncertainty theories. In the literature, various decision trees have been proposed to deal with

uncertain and incomplete data such as fuzzy decision trees [**?**], probabilistic decision trees [**?**], possibilistic decision trees [**?**, **?**, **?**] and belief decision trees [**?**, **?**, **?**]. The main advantage that makes the belief function theory very appealing over the other uncertainty theories, is its ability to express in a flexible way all kinds of information availability from full information to partial ignorance to total ignorance and also it allows to specify the degree of ignorance in a such situation. In this work, we focus our attention only on the belief decision trees approach developed by authors in [**?**] as an extension of the classical decision tree to cope with the uncertainty of the objects' classes and also allows to classify new objects described by uncertain attribute values [**?**]. In such a case, the uncertainty about the class' value is represented within the Transferable Belief Model (TBM), one interpretation of the belief function theory for dealing with partial or even total ignorance [**?**]. However, in several real data applications, uncertainty may appear in the attribute values [**?**]. For instance, in medicine, symptoms of patients may be partially uncertain. In this paper, we get inspired from the belief decision tree paradigm to handle data described by uncertain attribute values. Particulary, we tackle the case where the uncertainty occurs in both construction and classification phases. The reminder of this paper is organized as follows: Section 2 highlights the fundamental concepts of the belief function theory as interpreted by the TBM framework. In Section 3, we detail the building and the classification procedures of our new decision tree version. Section 4 is devoted to carrying out experiments on several real world databases. Finally, we draw our conclusion and our main future work directions in Section 5.

2 Belief function theory

In this Section, we briefly recall the fundamental concepts underlying the belief function theory as interpreted by the TBM [**?**].

Let us denote by Θ the frame of discernment including a finite non empty set of elementary events related to a given problem. The power set of Θ, denoted by 2^{Θ} is composed of all subsets of Θ.

The basic belief assignment (bba) expressing beliefs on the different subsets of Θ is a function $m : 2^{\Theta} \rightarrow$ [0,1] such that:

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

The quantity $m(A)$, also called basic belief mass (bbm), states the part of belief committed exactly to the event A. All subsets A in Θ such that $m(A) > 0$ are called focal elements.

Decision making within the TBM framework consists of selecting the most probable hypothesis for a given problem by transforming beliefs into probability measure called the pignistic probability and denoted by $BetP$. It is defined as follows:

$$BetP(A) = \sum_{B \subseteq \Theta} \frac{|A \cap B|}{|B|} \frac{m(B)}{1 - m(\emptyset)} \ \forall \ A \in \Theta \tag{2}$$

Let m_1 and m_2 be two bba's provided by fully reliable distinct information sources [**?**] and defined in the same frame of discernment Θ. The resulting bba using the conjunctive rule is defined by:

$$(m_1 \textcircled{\cap} m_2)(A) = \sum_{B,C \subseteq \Theta : B \cap C = A} m_1(B).m_2(C) \tag{3}$$

It is important to note that some cases require the combination of bba's defined on different frames of discernment. Let Θ_1 and Θ_2 be two frames of discernment, the vacuous extension of belief functions consists of extending Θ_1 and Θ_2 to a joint frame of discernment Θ 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:

$$\begin{aligned} &m^{\Theta_1 \uparrow \Theta}(A) = m_1(B) \; where \; A = B \times \Theta_2, \; B \subseteq \Theta_1 \\ &m^{\Theta_1 \uparrow \Theta}(A) = 0 \; otherwise \end{aligned} \tag{5}$$

3 Decision tree classifier for partially uncertain data

Authors in [**?**], have proposed what is called belief decision trees to handle real data applications described by known attribute values and uncertain class's value, particulary where the uncertainty is represented by belief functions within the TBM framework. However, for many real world applications, uncertainty may appear either in attribute values or in class value or in both attribute and class values. In this paper, we propose a novel decision tree version to tackle the case of uncertainty present only in attribute values for both construction and classification phases. Throughout this paper, we use the following notations:

- $\mathcal{T}$: a given training set composed by J objects I_j, $j = \{1,\ldots,J\}$.
- $\mathcal{L}$: a given testing of L objects O_l, $l = \{1,\ldots,L\}$.
- S: a subset of objects belonging to the training set $\mathcal{T}$.
- $C = \{C_1,\ldots,C_q\}$: represents the q possible classes of the classification problem.
- $A = \{A_1,\ldots,A_n\}$: the set of n attributes.
- Θ^{A_k}: represents the all possible values of an attribute $A_k \in A$, $k = \{1,\ldots,n\}$.
- $m^{\Theta^{A_k}}\{I_j\}(v)$: expresses the bbm assigned to the hypothesis that the actual attribute value of object I_j belongs to $v \subseteq \Theta^{A_k}$.

3.1 Decision tree parameters for handling uncertain attribute values

Four main parameters conducted to the construction of our proposed decision trees approach:

– **The attribute selection measure:** The attribute selection measure is relied on the entropy calculated from the average probability obtained from the set of objects in the node. To choose the most appropriate attribute, we propose the following steps:

1. Compute the average probability relative to each class, denoted by $Pr\{S\}(C_i)$, by taking into account the set of objects S. This function is obtained as follows:

$$Pr\{S\}(C_i) = \frac{1}{\sum_{I_j \in S} P_j^S} \sum_{I_j \in S} P_j^S \gamma_{ij} \tag{6}$$

   where γ_{ij} equals 1 if the object I_j belongs to the class C_i, 0 otherwise and P_j^S corresponds to the probability of the object I_j to belong to the subset S. Assuming that the attributes are independent, the probability P_j^S will be equal to the product of the different pignistic probabilities induced from the attribute bba's corresponding to the object I_j and enabling I_j to belong to the node S.

2. Compute the entropy $Info(S)$ of the average probabilities in S which is set to:

$$Info(S) = -\sum_{i=1}^{q} Pr\{S\}(C_i) log_2 Pr\{S\}(C_i) \tag{7}$$

3. Select an attribute A_k. For each value $v \in \Theta^{A_k}$, define the subset $S_v^{A_k}$ composed with objects having v as a value. As the A_k values may be uncertain, $S_v^{A_k}$ will contain objects I_j such that their pignistic probability corresponding to the value v is as follows:

$$BetP^{\Theta^{A_k}}\{I_j\}(v) \neq 0 \tag{8}$$

4. Compute the average probability, denoted by $Pr\{S_v^{A_k}\}$, for objects in subset $S_v^{A_k}$, where $v \in \Theta^{A_k}$ and $A_k \in A$. It will be set as:

$$Pr\{S_v^{A_k}\}(C_i) = \frac{1}{\sum_{I_j \in S_v^{A_k}} P_j^{S_v^{A_k}}} \sum_{I_j \in S_v^{A_k}} P_j^{S_v^{A_k}} \gamma_{ij} \tag{9}$$

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

5. Compute $Info_{A_k}(S)$ as discussed by Quinlan [**?**], but using the probability distribution instead of the proportions. We get:

$$Info_{A_k}(S) = \sum_{v \in \Theta^{A_k}} \frac{|S_v^{A_k}|}{|S|} Info(S_v^{A_k}) \tag{10}$$

where $Info(S_v^{A_k})$ is calculated from Equation 7 using $Pr\{S_v^{A_k}\}$ and we define $|S| = \sum_{I_j \in S} P_j^S$ and $|S_v^{A_k}| = \sum_{I_j \in S_v^{A_k}} P_j^{S_v^{A_k}}$.

6. Compute the information gain yielded by the attribute A_k over the set of objects S such that:

$$Gain(S, A_k) = Info(S) - Info_{A_k}(S) \tag{11}$$

7. Compute the *Gain Ratio* relative to the attribute A_k by the use of the $SplitInfo$

$$GainRatio(S, A_k) = \frac{Gain(S, A_k)}{SplitInfo(S, A_k)} \tag{12}$$

where the $SplitInfo$ value is defined as follows:

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

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

– **Partitioning Strategy:** The partitioning strategy, also called the splitting strategy, consists of splitting the training set according to the attribute values. As we only deal with categorical attributes, we create an edge for each attribute value chosen as a decision node. Due to the uncertainty in the attribute values, after the partitioning step each training instance may belong to more than one subset with a probability of belonging calculated according to the pignistic probability of its attribute values.

– **Stopping criteria:** Four key strategies are suggested as stopping criteria:
  1. The treated node contains only one instance.
  2. The treated node contains instances belonging to the same class.
  3. There is no further attribute to test.
  4. The gain ratio of the remaining attributes are equal or less than zero.

– **Structure of leaves:** Leaves, in our proposed decision tree classifier, will be represented by a probability distribution over the set of classed computed from the probability of instances belonging to these leaves. This is justified by the fact that leaves may contain objects with different class values called heterogeneous leaves. Therefore, the probability of the leaf L relative to each class $C_i \in C$ is defined as follows:

$$Pr\{L\}(C_i) = \frac{1}{\sum_{I_j \in L} P_j^L} \sum_{I_j \in L} P_j^L \gamma_{ij} \tag{14}$$

where P_j^L is the probability of the instance I_j to belong to the leaf L.

3.2 Decision tree procedures to deal with uncertain attribute values

By analogy to the classical decision tree, our new decision tree version will be composed mainly of two procedures: the construction of the tree from data present uncertain attributes and the classification of new instances described by uncertain attribute values.

A. Construction procedure

Suppose that $\mathcal{T}$ is our training set composed by J objects characterized by n uncertain attributes $A = \{A_1, \ldots, A_n\}$ represented within the TBM framework. Objects of $\mathcal{L}$ may belong to the set of classes $C = \{C_1, \ldots, C_q\}$. The different steps of our building decision tree algorithm are described as follows:

1. Create the root node of the decision tree that contain all the training set objects.
2. Check if the node verify the stopping criteria presented previously.
   - If yes, declare it as a leaf node and compute its probability distribution.
   - If not, the attribute that has the highest *GainRatio* will be designed as the root of the decision tree related to the whole training set.
3. Perform the partitioning strategy by creating an edge for each attribute value chosen as a root. This partition leads to several training subsets.
4. Create a root node for each training subset.
5. Repeat the same process for each training subset from the step 2.
6. Stop when all nodes of the latter level are leaves.

B. Classification procedure

Once our decision tree classifier is constructed, it is possible to classify new objects of the testing set $\mathcal{L}$ described by uncertain attribute values [**?**]. As previously mentioned, the uncertainty about a such attribute values A_k relative to a new object to classify can be defined by a bba $m^{\Theta^{A_k}}$ expressing the part of beliefs committed exactly to the different values of this attribute. The bba $m^{\Theta^{A_k}}$ will be defined on the frame of discernment Θ^{A_k} including all the possible values of the attribute A_k. Let us denote by Θ^A the global frame of discernment relative to all the attributes. It is equal to the cross product of the different Θ^{A_k}:

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

Since an object is characterized by a set of combination of values where each one corresponds to an attribute, we have firstly to look for the joint bba representing beliefs on the different attribute values relative to the new object to be classified. To perform this goal just have to apply the following steps:

- Extend the different bba's $m^{\Theta^{A_k}}$ to the global frame of attributes Θ^A. Thus, we get the different bba's $m^{\Theta^{A_k}\uparrow\Theta^A}$.

– Combine the different extended bba's through the conjunctive rule of combination:

$$m^{\Theta^A} = \bigcirc\!\!\!\!\bigcirc_{k=1,\ldots,n} m^{\Theta^{A_k}\uparrow\Theta^A} \tag{16}$$

Once we have obtained the joint bba denoted by m^{Θ^A}, we consider individually the focal elements of this latter. Let x be a such focal element. The next step in our classification task consists of computing the probability distribution $Pr[x](C_i)(i = 1,\ldots,q)$. It is important to note that the computation of this latter depends on the subset x and more exactly on the focal elements of the bba m^{Θ^A}:

- If the treated focal element x is a singleton, then $Pr[x](C_i)$ is equal to the probability of the class C_i corresponding to the leaf to which the focal element is attached.

- If the focal element is not a singleton (some attributes have more than one value), then we have to explore all possible paths relative to this combination of values. Two possible cases may arise:

  * If all paths lead to the same leaf, then $Pr[x](C_i)$ is equal to the probability of the class C_i relative to this leaf.

  * If these paths lead to distinct leaves, then $Pr[x](C_i)$ is equal to the average probability of the class C_i relative to the different leaves.

- Finally, each test object's probability distribution over the set of classes will be computed as follows:

$$Pr_l(C_i) = \sum_{x\subseteq\Theta^A} m^{\Theta^A}(x) Pr[x](C_i) \;\forall\; C \in \{C_1,\ldots,C_q\} \text{ and } l = \{1,\ldots,L\} \tag{17}$$

The most probable class of the object O_l is the one having the highest probability Pr_l.

4 Experimentations

To evaluate the feasibility of our novel decision trees approach, we have carried our experiments on real categorical databases obtained from the UCI repository [**?**]. Due to the computational cost of our proposed approach, we have performed our experiments on several small databases. Table 1 provides a brief description of these data sets where #Instances, #Attributes and #Classes denote respectively the total number of instances, the total number of attributes and the total number of classes. It is important to note that our approach can also be

Table 1. Description of databases

Databases	#Instances	#Attributes	#Classes
Tic-Tac-Toe	958	9	2
Parkinsons	195	23	2
Balloons	16	4	2
Hayes-Roth	160	5	3
Balance	625	4	3
Lenses	24	4	3

applied in the case of numerical databases when applying some kinds of data preprocessing such as discretization, etc.

Let us remind that our purpose is to construct our decision tree classifier from datasets characterized by uncertain attribute values. Thus, we propose to include uncertainty in attribute values by tacking into consideration the original data sets and a degree of uncertainty P such that:

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

The degree of uncertainty P takes value in the interval [0,1]:

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

The performance of our novel decision tree paradigm when classifying new objects can be measured through several classification accuracies. In this work, we relied on:

- The PCC criterion that represents the percent of correct classification of objects belonging to the test set. It is computed as follows:

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

  The number of well classified instances corresponds to the number of test instances for which the most probable classes obtained through our proposed decision tree classifier are the same as the real ones.

- The distance criterion: the main idea underling this criterion is to perform a comparison between a test instance's probability distribution over the set of classes and its real class. It is set as follows:

$$\begin{aligned} DistanceCriterion_j &= Distance(Pr_j(C_i), C(I_j)) \\ &= \sum_{i=1}^{q} (Pr_j(C_i) - \gamma_{ij})^2 \end{aligned} \tag{20}$$

where $C(I_j)$ corresponds to the real class of the test instance I_j and γ_{ij} equals 1 when $C(I_j) = C_i$ and 0 otherwise.

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.

We run our proposed classifier using the 10-folds cross validation technique that randomly split the original data set into 10 equal sized subsets. Of the 10 subsets, a single subset is used as a test data and the remaining subsets are used as training data. The cross-validation process is then repeated 10 times where each subset is used exactly once as a test set. Our experimental results in terms of classification accuracy and distance are depicted in Figure 1 and Figure 2 for the different mentioned databases.

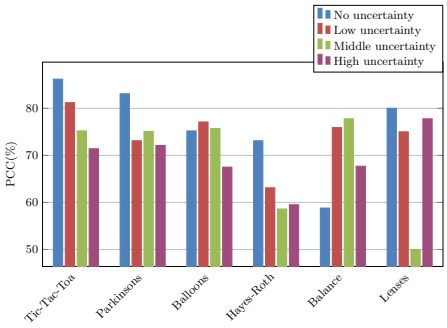

Fig. 1. PCC results

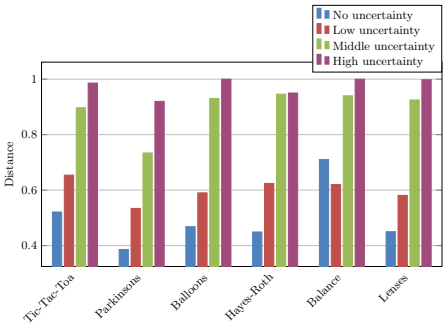

Fig. 2. Distance results

From the results given in Figure 1, we can remark that our proposed decision tree classifier has yielded good classification accuracy for the different uncertainty levels for the different databases. For instance, for Balloons database, we have 75.2%, 77.1%, 75.7% and 67.5% as PCCs relative respectively to no, low, middle and high uncertainties. Concerning the distance criterion, from Figure 2, we deduce that our classifier has given interesting results in term of distance criterion. In fact, all distance values belong to the closed interval [0.386,1]. Mostly, the distance increases with the increasing of the uncertainty degree. For example, the distance results relative to Balloons database are 0.46, 0.59, 0.93 and 1 for respectively no, low, middle and high uncertainties. This interpretation is available for the major remaining databases.

5 Conclusion

Tackling classification problem in the case of uncertainty present in attribute values remains a challenging task but currently very under-studied. Thus, in

this paper, we have proposed a new decision tree classifier to handle uncertainty present in the attribute values. Since we have obtained promising results, time complexity is still a critical problem, especially for large or even medium sized databases. So, as a future work we look forward reducing time complexity. We intend also to apply a pruning technique to reduce the dimensionality space and improve the classification accuracy.

References

[1] D. N. A. Asuncion. UCI machine learning repository, 2007.

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

[3] Z. Elouedi, K. Mellouli, and P. Smets. Classification with belief decision trees. In *Artificial Intelligence: Methodology, Systems, and Applications*, pages 80–90. Springer, 2000.

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

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

[6] I. Jenhani, N. B. Amor, and Z. Elouedi. Decision trees as possibilistic classifiers. *International Journal of Approximate Reasoning*, 48(3):784–807, 2008.

[7] I. Jenhani, Z. Elouedi, N. B. Amor, and K. Mellouli. Qualitative inference in possibilistic option decision trees. In *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, pages 944–955. Springer, 2005.

[8] J. R. Quinlan. Decision trees as probabilistic classifiers. In *4th international machine learning*, pages 31–37, 1897.

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

[10] J. R. Quinlan. *C4. 5: programs for machine learning*. Elsevier, 2014.

[11] A. Samet, E. Lefèvre, and S. B. Yahia. Evidential data mining: precise support and confidence. *Journal of Intelligent Information Systems*, pages 1–29, 2016.

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

[13] P. Smets. The transferable belief model for quantified belief representation. In *Quantified Representation of Uncertainty and Imprecision*, pages 267–301. Springer, 1998.

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

[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. On aggregating belief decision trees. *Information fusion*, 5(3):179–188, 2004.

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