Available online at www.sciencedirect.com
ScienceDirect
International Journal of Approximate Reasoning
48 (2008) 784–807
INTERNATIONAL JOURNAL OF APPROXIMATE REASONING
www.elsevier.com/locate/ijar

Decision trees as possibilistic classifiers

Ilyes Jenhani *, Nahla Ben Amor, Zied Elouedi

LARODEC, Institut Supérieur de Gestion, Tunis, Tunisia

Received 25 November 2006; received in revised form 25 October 2007; accepted 14 December 2007
Available online 20 January 2008

Abstract

This paper addresses the classification problem with imperfect data. More precisely, it extends standard decision trees to handle uncertainty in both building and classification procedures. Uncertainty here is represented by means of possibility distributions. The first part investigates the issue of building decision trees from data with uncertain class values by developing a non-specificity based gain ratio as the attribute selection measure which, in our case, is more appropriate than the standard gain ratio based on Shannon entropy. The proposed non-specificity based possibilistic decision tree (NS-PDT) approach is then extended by considering another kind of uncertainty inherent in the building procedure. The extended approach so-called non-specificity based possibilistic option decision tree (NS-PODT) offers a more flexible building procedure by allowing the selection of more than one attribute in each node. The second part addresses the classification phase. More specifically, it investigates the issue of predicting the class value of new instances presented with certain and/or uncertain attribute values. Finally, we have developed a possibilistic decision tree toolbox (PD2T) in order to show the feasibility of the proposed approach.

Keywords: Classification; Decision trees; Possibility theory; Non-specificity

1. Introduction

Classification represents an important task in machine learning and data mining applications. It consists in (1) inducing a classifier from a set of historical examples (training set) with known class values and then (2) using the induced classifier to predict the class value (the category) of new objects given known the values of their attributes (features).

This task is ensured by a panoply of techniques: statistical techniques (e.g. discriminant analysis, etc.) and artificial intelligence based techniques (e.g. artificial neural networks, *k*-nearest neighbors, Bayesian networks, decision trees, etc.). The latter, namely, decision trees, are considered as one of the most popular classification techniques [49]. They are able to represent knowledge in a flexible and easy form which justifies their use in decision support systems, intrusion detection systems, medical diagnosis, etc.

* Corresponding author.
E-mail addresses: ilyes.j@lycos.com (I. Jenhani), nahla.benamor@gmx.fr (N.B. Amor), zied.elouedi@gmx.fr (Z. Elouedi).

0888-613X/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.ijar.2007.12.002

In many real-world problems, classes of examples in the training set may be partially defined and even missing. For example, for some instances, an expert may be unable to give the exact class value. A doctor who cannot specify the exact disease of a patient, a banker who cannot decide whether to give or not a loan for a client, a network administrator who is not able to decide about the exact signature of a given connection, etc. Hence, in these different examples, the expert can provide imprecise or uncertain classifications expressed in the form of a ranking on the possible classes. Ignoring the uncertainty may affect the classification results and even produce erroneous decisions. Consequently, ordinary classification techniques such as decision trees should be adequately adapted to take care of this problem.

Our idea is to treat different levels of uncertainty using possibility theory which is a non-classical theory of uncertainty proposed by Zadeh [53] and developed by Dubois and Prade [14]. More precisely, we will handle training instances whose class labels are given in the form of possibility distributions. We also adapt the attribute selection measure, used in the building phase, to the possibilistic framework by using a non-specificity based criterion instead of the Shannon entropy [44]. Such possibilistic decision tree will be referred to by NS-PDT.

In addition to the uncertainty that might characterize training data, another source of uncertainty is hidden in the building procedure of decision trees [27]. In fact, the core of the building procedure is based on an heuristic function, namely, the attribute selection measure which enables us to choose the "most" informative attribute at each decision node of the tree under construction. Hence, we collide with a kind of uncertainty which is related to the choice of an attribute at a given decision node.

In a previous work [27], we developed what we have called *possibilistic option decision trees* (PODT) where each decision node can be split according to more than one attribute (using multiple attribute-value tests, or "options"). Different options are quantified via possibility distributions. In this paper, we will extend the NS-PDT approach to deal with uncertainty within the attribute selection step, this extension will be referred to by NS-PODT.

Once an NS-PODT is constructed, it will be used to classify new instances. In this work, we have considered the case where all attributes are well known and the case where some or even all of them have uncertain values (e.g. imprecise or missing attribute values). Such situation can appear, for instance, when using sensors to provide attribute values of new instances. Uncertainty in the classification phase will also be modeled in the possibilistic framework.

An alternative *possibilistic decision tree* induction method was proposed by Borgelt et al. in [4]. Nevertheless, contrary to our approach, the proposed classifier is not able to treat uncertain instances and possibilities appear in the building phase when frequency distributions are taken as possibility distributions used in order to define a possibilistic attribute selection measure. Another possibilistic induction method was proposed by Hüllermeir [23]. In his work, the author applied a possibilistic branching on Lazy decision trees [18]. A work by Ben Amor et al. [2] dealt with possibilistic uncertainty, only within the classification phase of decision tree technique.

Other non-standard decision trees were proposed. Namely, *fuzzy decision trees* [26,34,38–40,52] which blend decision trees with fuzzy set tools to manage fuzzy information (attribute and class values are vaguely expressed with linguistic fuzzy terms) or to fuzzify the crisp rules extracted from a standard decision tree. *Probabilistic decision trees* [7,42] and *belief decision trees* [10,16,46,47] were also proposed to deal with uncertainty in data represented, respectively, by means of probability distributions and basic belief assignments. A deep analysis of these proposals with respect to our approach will be presented in Section 7.

The paper is organized as follows: Section 2 starts by giving the necessary background concerning the decision tree classification technique. Section 3 recalls some aspects of possibility theory as well as the concept of non-specificity. The characteristics and parameters of the non-specificity based possibilistic decision tree approach (NS-PDT) are then defined in Section 4. Section 5 proposes an extension of NS-PDT, namely the NS-PODT approach. This section defines the building procedure, then, it describes the method which we propose for the classification of certain and/or uncertain instances within the NS-PODT approach. Section 6 presents and analyzes experimental results carried out on modified versions of commonly used data sets obtained from the U.C.I. machine learning repository [37]. Before concluding, a summary of related works is provided in Section 7.

2. Decision trees

A decision tree is a flow-chart-like hierarchical tree structure which is composed of three basic elements: decision nodes corresponding to attributes, edges or branches which correspond to the different possible attribute values. The third component is leaves including objects that typically belong to the same class or that are very similar. Such representation allows us to induce decision rules that will be used to classify new instances. In fact, each path from the root to a leaf corresponds to a conjunction of test attributes and the tree is considered as a disjunction of these conjunctions.

The majority of decision trees is made up of two major procedures: the building (induction) and the classification (inference) procedures.

- *Building procedure*: Given a training set, building a decision tree is usually done by starting with an empty tree and selecting for each decision node the 'appropriate' test attribute using an attribute selection measure. The principle is to select the attribute that maximally diminish the mixture of classes between each training subset created by the test, thus, making easier the determination of object's classes. The process continues for each sub decision tree until reaching leaves and fixing their corresponding classes.
- *Classification procedure*: To classify a new instance, having only values of all its attributes, we start with the root of the constructed tree and follow the path corresponding to the observed value of the attribute in the interior node of the tree. This process is continued until a leaf is encountered. Finally, we use the associated label to obtain the predicted class value of the instance at hand.

Several algorithms for building decision trees have been developed. The most popular and applied ones are: **ID3** [41] and its successor **C4.5** "the state-of-the-art" algorithm developed by Quinlan [43]. We can also mention the **CART** algorithm of Breiman et al. [5].

Decision tree algorithms have many common components to be defined. These components are described as follows:

(a) *Attribute selection measure* generally based on information theory, serves as a criterion in choosing among a list of candidate attributes at each decision node, the attribute that generates partitions where objects are distributed less randomly, with the aim of constructing the smallest tree among those consistent with the data. The well-known measure used in the **C4.5** algorithm of Quinlan [43] is the gain ratio.

Given an attribute A_k, the information gain relative to A_k is defined as follows:

$$Gain(T, A_k) = E(T) - E_{A_k}(T), \tag{1}$$

where

$$E(T) = -\sum_{i=1}^{n} \frac{n(C_i, T)}{|T|} \log_2 \frac{n(C_i, T)}{|T|} \tag{2}$$

and

$$E_{A_k}(T) = \sum_{v \in D(A_k)} \frac{|T_v^{A_k}|}{|T|} E(T_v^{A_k}), \tag{3}$$

$n(C_i, T)$ denotes the number of objects in the training set T belonging to the class C_i, $D(A_k)$ denotes the finite domain of the attribute A_k and $|T_v^{A_k}|$ denotes the cardinality of the set of objects for which the attribute A_k has the value v. Note that $\frac{n(C_i,T)}{|T|}$ corresponds to the probability of the class C_i in T. Thus, $E(T)$ corresponds to the *Shannon entropy* [44] of the set T. This function will be more explained in Section 4.

The gain ratio is given by

$$Gr(T, A_k) = \frac{Gain(T, A_k)}{SplitInfo(T, A_k)}, \tag{4}$$

where *Split Info*(T, A_k) represents the potential information generated by dividing T into n subsets. It is given by

$$SplitInfo(T, A_k) = -\sum_{v \in D(A_k)} \frac{|T_v^{A_k}|}{|T|} \log_2 \frac{|T_v^{A_k}|}{|T|}. \quad (5)$$

(b) *Partitioning strategy* consisting in partitioning the training set according to all possible attribute values (for discrete attributes) which leads to the generation of one partition for each possible value of the selected attribute. For continuous attributes, we need a discretization step. Different discretization strategies exist in the literature [17] (e.g. the simple discretization, etc.). We do not detail them because, in this paper, we only deal with discrete attributes.

(c) *Stopping criteria* stopping the partitioning process. Generally, we stop the partitioning if all the remaining objects belong to only one class, then the node is declared as a leaf labeled with this class value. We, also, stop growing the tree if there is no further attribute to test. In this case, we take the majority class as the leaf's label.

3. Basics of possibility theory

Possibility theory represents a non-classical theory (distinct from probability theory) which offers a natural and simple model that deals with both uncertain and imprecise information. In this section, we will give a brief recalling on possibility theory (for more details see [13]). Then, we will focus on the source of uncertainty relative to possibility theory, namely, the *non-specificity*.

3.1. Basic elements

Given $\Omega = \{\omega_1, \omega_2, \ldots, \omega_n\}$, the universe of discourse, the basic concept of possibility theory is the notion of *possibility distribution* denoted by π and which corresponds to a function which associates to each element ω_i of the universe of discourse Ω a value from a bounded and linearly ordered valuation set $(L, <)$. This value is called a *possibility degree*: it encodes our knowledge, denoted by u, on the real world. Note that, in possibility theory, the scale can be numerical (e.g. $L = [0, 1]$): in this case we have numerical possibility degrees from the interval $[0, 1]$ and hence we are dealing with the quantitative setting of the theory. In the qualitative setting, it is the ordering between the different possible values that is important but the numerical values themselves have no sense.

By convention, $\pi(\omega_i) = 1$ means that it is fully possible that $\omega_i = u$ is the real world, $\pi(\omega_i) = 0$ means that $\omega_i = u$ cannot be the real world (is impossible), and $\pi(\omega_i) > \pi(\omega_j)$ means that $\omega_i = u$ is preferred to $\omega_j = u$ (or is more plausible).

A possibility distribution π is said to be *normalized* if there exists at least one state ω_k which is totally possible (i.e. $\pi(\omega_k) = 1$). In this paper, we only deal with normalized possibility distributions.

In the possibilistic framework, extreme cases of knowledge are presented by

- *complete knowledge*: $\exists \omega_0$, $\pi(\omega_0) = 1$ *and* $\pi(\omega) = 0$ $\forall \omega \neq \omega_0$.
- *total ignorance*: $\pi(\omega) = 1 \forall \omega$ (all values in Ω are possible).

Possibility theory is based on two dual measures: *possibility* and *necessity* measures. Given the universe of discourse Ω and a possibility distribution π on Ω, the corresponding *possibility* and *necessity* measures of any event $\varphi \subseteq \Omega$ are, respectively, determined by the formulas:

$$\Pi(\varphi) = \max_{\omega \in \varphi} \pi(\omega), \quad (6)$$

$$N(\varphi) = \min_{\omega \notin \varphi} (1 - \pi(\omega)). \quad (7)$$

$\Pi(\varphi)$ evaluates at which level φ is *consistent* with our knowledge represented by π while $N(\varphi)$ evaluates at which level φ is *certainly* implied by our knowledge represented by π. Note that for any $\varphi \subseteq \Omega$: $N(\varphi) = 1 - \Pi(\bar{\varphi})$.

As we can see from Eq. (6), the basic axiom of possibility theory is the *maximum* operator. Hence, the possibility of the disjunction of two events φ_1 and φ_2 is the maximum of the respective possibility of the individual events.

$$\Pi(\varphi_1 \vee \varphi_2) = \max(\Pi(\varphi_1), \Pi(\varphi_2)). \tag{8}$$

Suppose that a possibility distribution π is provided by a given source (e.g. expert, sensor) and suppose that the degree of certainty that this source is reliable is given by β, then π can be updated into [13]:

$$\pi' = \max(\pi, 1 - \beta). \tag{9}$$

Note that when $\beta = 1$ (fully reliable source), $\pi' = \pi$ and in the case of absolutely unreliable source ($\beta = 0$), $\forall\omega,\pi'(\omega) = 1$ (total ignorance). Eq. (9) represents a form of **discounting** of a given possibility distribution.

3.2. Non-specificity

As mentioned by Higashi and Klir [22], possibility theory deals with a source of uncertainty called: *non-specificity*. This type of uncertainty is manifested in our inability to distinguish which of several possible alternatives is the true one in a particular situation. The larger the set of possible alternatives is, the larger is the non-specificity. More precisely, non-specificity is connected with sizes (cardinalities) of relevant sets of alternatives [30].

Given two possibility distributions π and π' both on Ω, π is said to be *more specific than* π' if and only if, for each $\omega \in \Omega$, $\pi(\omega) \leqslant \pi'(\omega)$ [51]. Clearly, the more specific π, the more informative it is.

Thus, in some decision-making situations, one need to measure the amount of uncertainty inherent in each given possibility distribution in order to decide which one is the most informative. The first measure of non-specificity was proposed by Hartley [21] for classical set theory which represents the simplest means by which we can express uncertainty. The Hartley function for a subset A of a universal finite set X is given by

$$H(A) = \log_2|A|. \tag{10}$$

Here, A corresponds to the smallest subset of X such that we are certain that the actual state is in A.

The majority of non-specificity measures proposed for other uncertainty frameworks (e.g. evidence theory, fuzzy set theory, possibility theory, etc.) represents a generalization of Hartley function. For instance, for the possibilistic setting, the measure of non-specificity, called *U-uncertainty* and proposed by Higashi and Klir [22], has the form:

$$U : \mathscr{R} \rightarrow \mathbf{IR}^+,$$

where $\mathscr{R}$ denotes the set of all finite and ordered possibility distributions. Given an ordered possibility distribution $\pi = \langle \pi_{(1)}, \pi_{(2)}, \ldots, \pi_{(n)} \rangle$ such that $1 = \pi_{(1)} \geqslant \pi_{(2)} \geqslant \cdots \geqslant \pi_{(n)}$, the *U-uncertainty* of π, is given by the formula:

$$U(\pi) = \sum_{i=1}^{n} (\pi_{(i)} - \pi_{(i+1)}) \log_2 i, \tag{11}$$

where $\pi_{(n+1)} = 0$ by convention [29]. Note that the range of U is $[0, \log_2 n]$. $U(\pi) = 0$ is obtained for the case of complete knowledge (no uncertainty) and $U(\pi) = \log_2 n$ is reached for the case of total ignorance.

4. Non-specificity based possibilistic decision trees (NS-PDT)

A non-specificity based possibilistic decision tree is a decision tree with the same representation of an ordinary decision tree, i.e., it is composed of *decision nodes* for testing attributes, *branches* specifying attribute values and *leaves* dealing with classes of the training set.

In supervised learning, more specifically, in classification problems, we need a set of historical examples with known classes, called the training set, from which we will train a classifier (e.g. a decision tree). Then, this classifier will be used to predict the class value of each new object given known its attributes' values.

4.1. Imperfection in classification problems

As models of the real world, databases, or more specifically, training sets are often permeated with forms of imperfections, including imprecision and uncertainty. The topic of imperfect databases is gaining more and more attention the last years [33,35] since commercial database management systems are not able to deal with such kind of information. Now, we ask what is imperfect in a training set and why is it imperfect?

Imperfection in a training set may affect attribute values as well as class values, for instance, the *departure_time* of a flight, the *temperature* of a patient, the *property_value* of a client asking for a loan. Examples of imperfect class values include the exact type of an attack in an intrusion detection system, the exact cancer class of a patient in cancer diagnosis applications, the exact location or type of a detected aerial engine in military applications, etc.

Another interesting real example emphasizing the problem of having imprecise class labels is the one given in [11]. It consists in detecting certain transient phenomena (e.g. *k*-complexes and delta waves) in electroencephalogram (EEG) data. Such phenomena are usually difficult to detect, hence doctors are not always able to recognize them with full certainty. Consequently, it may be more easy for the doctor to assess the possibility that certain phenomena are present in the data.

These imperfections might result from using unreliable information sources, such as faulty reading instruments, or input forms that have been filled out incorrectly (intentionally or inadvertently). In other cases, imperfection is a result of system errors, including transmission noise, network latency for sensor networks applications, delays in processing update transactions, etc.

In a learning process, we should never reject or ignore such information (by affecting the *null* value to such information) despite of its imperfection. On the contrary, we should benefit from the maximum amount of information which should be handled carefully else the learnt model could be inaccurate or even incorrect. The first paper dealing with learning from uncertain data is attributed to Denoeux [8]. The author has extended the well-known *k*-nearest neighbor classifier to handle uncertain data by using belief function theory.

In this work, we only deal with imprecise class labels in the training set. Instead of rejecting instances having imprecise class labels or adding a *null* class value to such instances, we used a convenient mathematical model to deal with such kind of imperfection, namely possibility theory [13,53]. More formally, a possibility degree will be assigned to each possible class value indicating the possibility that the instance belongs to a given class [11,54] instead of using a simple set of disjunctive values with equal weights. These possibility degrees can be obtained from direct expert's elicitation, i.e., each expert is asked to quantify by a real number between 0 and 1 the possibility that a training instance belongs to each one of the different classes of the problem. Possibilistic class labels may also be obtained from an empirical distribution of expert opinions using possibilistic histograms [12].

The question that arises is: how to induce decision trees from training instances, classes of which are presented by means of possibility distributions?

4.2. Building procedure

Standard building procedure (see Section 2) starts with an empty tree. The first step consists in selecting the most informative attribute, i.e., the attribute that, if assigned to the decision node at hand, will produce the least conflicting training subsets towards instances' classes. Note that measuring the conflict of a training (sub)set comes down to measure the conflict of the probability distribution on the different classes of the instances belonging to that set.

For instance, suppose that, in a 2-class problem, we have a training subset T_1 with five instances: two instances having class C_1 and the remaining three instances are labeled with class C_2. Then, the probabilities of each class in T_1 are respectively 2/5 and 3/5. Thus, the probability distribution characterizing T_1 is $p = (0.4, 0.6)$. Finally, we need to measure the amount of conflict in p, which represents the only source of "uncertainty" in probability theory [30]. The well-established measure of conflict in probability theory is the Shannon entropy given in Eq. (2). Hence, $E(T_1) = E(p) = -0.4\log_2(0.4) - 0.6\log_2(0.6) = 0.971$. Note that for a uniform probability distribution on X, i.e., $p(x) = 1/|X|\ \forall x \in X$ (the case of total conflict), $E(p) = \log_2|X|$. However, in the case of total certainty, i.e., $p(x) = 1$ for some $x \in X$, $E(p) = 0$.

In our case, instead of precise classes, we have possibility distributions over different classes of each training instance. Hence, each training (sub)set will be characterized by the (sub)set of possibility distributions relative to the instances of that (sub)set. Therefore, in order to discriminate between two or n sets, one should measure the amount of uncertainty of such sets and then select the attribute generating the least uncertain subsets.

4.2.1. Measuring uncertainty of sets of possibility distributions

Harmanec said in [20] "*Before we can measure uncertainty or information, we have to be clear what exactly we are trying to measure...*". In other words, one should, first, determine the source(s) of uncertainty ingrained in our mathematical model, then use the suitable measure of uncertainty relatively to each source.

Let us return to our problem: we want to discriminate between different sets of possibility distributions. The first idea that comes up is to merge the different possibility distributions of each set using the well-known fusion operators relative to possibility theory, namely, the t-norms and the t-conorms [14]. In this way, we obtain a single possibility distribution for each set and we can measure their different non-specificities (using Eq. (11)) in order to choose the most specific one, i.e., the least non-specific one.

This procedure is problematic. In fact, in the decision tree context, in each node, we have possibility distributions of distinct training instances reaching that node. These instances have some common attribute values (those values labeling edges of the path leading to that node) and the remaining attributes may have different values. So, it is clear that we cannot merge possibility distributions which are not dealing with the same "object": a necessary condition for information fusion problems. We generally use the well-known fusion operators relative to possibility theory, namely, the t-norms and the t-conorms [14] when we have to merge *different* information (e.g. possibility distributions) provided by *different* or even a unique source (e.g. a sensor giving information at different times, etc.) on the *same* observed object.

Since fusion is not the appropriate tool in this context, the solution that we propose is the following: for each set containing m possibility distributions, we will induce a representative possibility distribution of that set (π_{Rep}), that is, a possibility distribution that represents the proportion of the different possibility degrees of the different values (class values). This possibility distribution is obtained via the arithmetic mean of $\pi^{j=1,\ldots,m}$ possibility distributions [3] and it is given by

$$\pi_{\text{AM}}(\omega_i) = \frac{1}{m}\left(\sum_{j=1}^{m} \pi^j(\omega_i)\right). \tag{12}$$

Then, we should normalize π_{AM} to obtain:

$$\pi_{\text{Rep}}(\omega_i) = \frac{\pi_{\text{AM}}(\omega_i)}{\max_{i=1}^{|\Omega|}\{\pi_{\text{AM}}(\omega_i)\}}. \tag{13}$$

Finally, we can measure the non-specificity of π_{Rep} using Eq. (11) and hence, discriminate between different sets of possibility distributions.

4.2.2. Components of NS-PDT

As mentioned in Section 2, ordinary decision tree algorithms are made up of three basic components, namely, (a) attribute selection measure, (b) partitioning strategy and (c) stopping criteria. Let us define these components for the NS-PDT approach.

4.2.2.1. Attribute selection measure. For our attribute selection measure, we were inspired by the well-known standard attribute selection measure, namely, the gain ratio criterion proposed by Quinlan [43]. Indeed, as explained above, this measure, which is essentially based on Shannon entropy, cannot be applied in the case of possibilistic labels. Hence, we have used the counterpart of the probabilistic Shannon entropy in possibility theory, namely, the non-specificity measure which also satisfies a set of mathematical properties: additivity, expansibility, symmetry, branching, monotonicity, etc. [30].

Roughly speaking, standard attribute selection measures (e.g. gain ratio) will select the attribute giving more homogeneous partitions, i.e., giving less random partitions (i.e. with less entropy values). Likewise, in

our possibilistic case, we will select the attribute resulting in less imprecise partitions (i.e. with less non-specificity values).

Given a training set T in which instances's classes are presented in the form of possibility distributions over the different possible class values and given the set of attributes, the non-specificity gain (NSG) of an attribute A_k is defined by

$$NSG(T, A_k) = U(\pi^T_{\text{Rep}}) - U_{A_k}(\pi^T_{\text{Rep}}), \tag{14}$$

where $U(\pi^T_{\text{Rep}})$ denotes the non-specificity of the possibility distribution representing the set of possibility distributions in the set T.

And

$$U_{A_k}(\pi^T_{\text{Rep}}) = \frac{1}{|D(A_k)|} \sum_{v \in D(A_k)} U(\pi^{T^{A_k}_v}_{\text{Rep}}). \tag{15}$$

$NSG(T, A_k)$ assesses the amount of "information precision" obtained after splitting our training set according to the attribute A_k. Note that $NSG(T, A_k) \in [-\log_2(n), \log_2(n)]$ where n denotes the number of classes of the problem ($n = |\Omega|$). Note the following particular values of $NSG(T, A_k)$:

- $NSG(T, A_k) = -\log_2(n)$: means that splitting according to A_k will result in the maximum loss in precision, i.e., resulting subsets are less precise (in average) than the starting training partition.
- $NSG(T, A_k) = 0$: means that splitting according to A_k will result in any loss nor any gain in precision.
- $NSG(T, A_k) = \log_2(n)$: means that splitting according to A_k will result in the maximum gain in precision, i.e., from a maximally imprecise set, we obtain maximally precise subsets.

Similarly to the C4.5 algorithm, in order to ovoid bias for attributes with many values, we will divide $NSG(T, A_k)$ by $SplitInfo(T, A_k)$ (see Eq. (5)):

$$NSGr(T, A_k) = \frac{NSG(T, A_k)}{SplitInfo(T, A_k)}. \tag{16}$$

Obviously, the attribute maximizing $NSGr$ (non-specificity gain ratio) will be assigned to the decision node at hand.

Note that our approach covers the special case of certain training data, i.e., training data which are labeled by certain possibility distributions. In this case, π_{AM} will correspond to the frequency distribution ($freq$) of the different classes in a training partition. Consequently, the possibility distribution π_{Rep} will represent a kind of normalized frequency $Nfreq$ (a division by the maximum frequency). So an important question is: how does $U(Nfreq)$ relate to $E(freq)$? (where U and E denote, respectively, the *U-uncertainty* and the *Entropy* measures).

To respond to this question, we should note the following cases:

- If all instances in a training partition have the same class. This case characterizes a situation of complete knowledge. Consequently, the possibility distribution π_{AM} (which is normalized by nature) will be equivalent to the frequency distribution $freq(\pi_{\text{AM}} = freq = Nfreq)$. For instance, if we are dealing with a training data with four classes, we will have $\pi_{\text{AM}} = Nfreq = freq = (1, 0, 0, 0)$. Thus, we obtain $U(Nfreq) = E(freq) = 0$.
- If a partition contains exactly the same proportion of classes. This case characterizes a situation of total ignorance. Again, the possibility distribution π_{AM} (which is not normalized here) will be equivalent to the frequency distribution $freq(\pi_{\text{AM}} = freq)$. Normalizing π_{AM} will result in a fully non-specific possibility distribution $Nfreq$. For instance, if we are dealing with a training data with four classes, we will have $freq = (0.25, 0.25, 0.25, 0.25)$ and $Nfreq = (1, 1, 1, 1)$. Again, we obtain the equality: $U(Nfreq) = E(freq) = \log_2(4) = 2$.
- If a partition is neither pure nor fully non-specific, $Nfreq$ will be different to $freq$ and $U(Nfreq) \neq E(freq)$. A more relevant conclusion is: $E(freq1) > E(freq2)$ does not always imply that $U(Nfreq1) > U(Nferq2)$. In fact, let $freq1 = (0.5, 0.2, 0.2, 0.1)$ and $freq2 = (0.4, 0.3, 0.3, 0)$. Thus, $Nfreq1 = (1, 0.4, 0.4, 0.2)$ and $Nfreq2 =$

$(1, 0.75, 0.75, 0)$. This example shows that $E(freq1) = 1.76 > E(freq2) = 1.57$ but $U(Nfreq1) = 0.717 < U(Nfreq2) = 1.18$.

Clearly, the last case, which is the most occurring in a decision tree building procedure, shows that our measure and the entropy based measure do not always select the same attribute even in the case where the training set is labeled by certain possibility distributions.

4.2.2.2. Partitioning strategy. Once an attribute is selected at a given decision node and since we only deal with discrete attributes, the partitioning strategy will be the same as with ordinary decision trees, i.e., for each value of the selected attribute, an edge labeled with that value is added. The process continues, recursively, for each generated training partition T_p as described in Section 2.

4.2.2.3. Stopping criteria. Since our approach is dealing with training instances, classes of which are characterized by possibility distributions, the stopping criteria, mentioned in Section 2 cannot be directly applied and should be adapted to such a situation. We propose the following five cases for which we should stop growing the tree:

For each generated training partition T_p:

1. If there is no further attribute to test, we declare a leaf labeled by $\pi_{\text{Rep}}^{T_p}$: the possibility distribution representing the set of possibility distributions in T_p.
2. Else, if NSGr $\leqslant 0$ (no information is gained). On the contrary, when continuing splitting, we will lose information. In this case, the leaf will be labeled by $\pi_{\text{Rep}}^{T_p}$.
3. Else, if the non-specificity of the possibility distribution representing the training partition T_p equals to 1 and $\pi_{\text{AM}}^{T_p} \equiv \pi_{\text{Rep}}^{T_p}$ $(U(\pi_{\text{AM}}^{T_p}) = U(\pi_{\text{Rep}}^{T_p}) = 1)$, i.e., the partition contains only fully non-specific possibility distributions (total ignorance). In this case, continuing the partitioning is useless. Hence, we declare a leaf labeled by $\pi_{\text{Rep}}^{T_p}$.
4. Else, if the non-specificity of the possibility distribution representing the training partition T_p equals to 0 $(U(\pi_{\text{Rep}}^{T_p}) = 0)$, i.e., all instances in the training partition have the same class (complete knowledge). In this case, we declare a leaf labeled by the fully specific possibility distribution $\pi_{\text{Rep}}^{T_p}$.
5. Else, if the non-specificity of the possibility distribution representing the training partition T_p is less or equal to a pre-computed threshold $AvgU$ $(U(\pi_{\text{Rep}}^{T_p}) <= AvgU)$, i.e., the training partition is enough pure. Thus, we declare a leaf labeled by $\pi_{\text{Rep}}^{T_p}$. $AvgU$ represents the average of non-specificity measures of each possibility distribution of each instance of the whole training set T:

$$AvgU = \frac{1}{|T|} \sum_{r=1}^{|T|} U(\pi^r), \tag{17}$$

where $|T|$ denotes the number of instances of the training set and π^r denotes the possibility distribution on the classes of the rth instance in T. Note that the value of $AvgU$ ranges in the interval $[0, \log_2|C|]$ where C corresponds to the set of classes of the problem.

It is important to mention that, when stopping criterion 3 or 4 are satisfied, the value of the NSGr criterion will be equal to 0, i.e., no information gain is obtained. We separated these criteria just to show the two particular cases of total ignorance and complete knowledge.

Example 1. Let us use a modified version of the golf data set [37] to illustrate the induction of a non-specificity based possibilistic decision tree (NS-PDT). Let T be the training set composed of fourteen instances $I_{i=1,\ldots,14}$ which are characterized by four attributes:

- *Outlook*: sunny or overcast or rainy.
- *Temp*: hot or mild or cool.
- *Humidity*: high or normal.
- *Wind*: weak or strong.

Table 1
Training set

	Outlook	Temp	Humidity	Wind	C_1	C_2
I_1	Sunny	Hot	High	Weak	0.2	1
I_2	Sunny	Hot	High	Strong	0.4	1
I_3	Overcast	Hot	High	Weak	1	0.7
I_4	Rainy	Mild	High	Weak	1	0
I_5	Rainy	Cool	Normal	Weak	1	0.8
I_6	Rainy	Cool	Normal	Strong	0.4	1
I_7	Overcast	Cool	Normal	Strong	1	0.9
I_8	Sunny	Mild	High	Weak	0.3	1
I_9	Sunny	Cool	Normal	Weak	1	0.3
I_{10}	Rainy	mild	Normal	Weak	1	0
I_{11}	Sunny	Mild	Normal	Strong	1	0.2
I_{12}	Overcast	Mild	High	Strong	1	0
I_{13}	Overcast	Hot	Normal	Weak	1	0.3
I_{14}	Rainy	Mild	High	Strong	0	1

Two classes are possible either, C_1 (play) or C_2 (do not play). We have added uncertainty to T in an artificial manner: a possibility distribution was given for each class of each instance of T. The training set T is given by Table 1

Let us, first, compute $AvgU$ (using Eq. (17)):

$$AvgU = \frac{1}{|T|}\sum_{r=1}^{|T|} U(\pi_r) = \frac{1}{14}(0.2+0.4+0.7+0+0.8+0.4+0.9+0.3+0.3+0+0.2+0+0.3+0) = 0.321.$$

Let us show a detailed computation of the non-specificity gain ratio of only one attribute, namely, the "Humidity" attribute. One should first determine π^T_{Rep}: From Table 1, using Eq. (12), we obtain:

$$\pi^T_{\text{AM}} = \left(\frac{(0.2+0.4+1+1+1+0.4+1+0.3+1+1+1+1+1+0)}{14}, \times \frac{(1+1+0.7+0+0.8+1+0.9+1+0.3+0+0.2+0+0.3+1)}{14}\right) = [0.73, 0.58].$$

We normalize (using Eq. (13)) to obtain:

$$\pi^T_{\text{Rep}} = \left[\frac{0.73}{0.73}, \frac{0.58}{0.73}\right] = [1, 0.79] \Rightarrow U(\pi^T_{\text{Rep}}) = 0.79.$$

Let us now determine $\pi^{T^{\text{Humidity}}_{\text{high}}}_{\text{Rep}}$ and $\pi^{T^{\text{Humidity}}_{\text{normal}}}_{\text{Rep}}$:

$$\pi^{T^{\text{Humidity}}_{\text{high}}}_{\text{AM}} = \left[\frac{(0.2+0.4+1+1+0.3+1+0)}{7}, \frac{(1+1+0.7+0+1+0+1)}{7}\right]$$

$$= [0.557, 0.671] \Rightarrow \pi^{T^{\text{Humidity}}_{\text{high}}}_{\text{Rep}} = [0.83, 1]$$

$$\Rightarrow U(\pi^{T^{\text{Humidity}}_{\text{high}}}_{\text{Rep}}) = 0.83$$

$$\pi^{T^{\text{Humidity}}_{\text{normal}}}_{\text{AM}} = \left[\frac{(1+0.4+1+1+1+1+1)}{7}, \frac{(0.8+1+0.9+0.3+0+0.2+0.3)}{7}\right]$$

$$= [0.914, 0.5] \Rightarrow \pi^{T^{\text{Humidity}}_{\text{normal}}}_{\text{Rep}} = [1, 0.547] \Rightarrow U(\pi^{T^{\text{Humidity}}_{\text{normal}}}_{\text{Rep}}) = 0.547$$

$$\Rightarrow U_{Humidity}(\pi^{T}_{\text{Rep}}) = \frac{1}{2}(0.83 + 0.547) = 0.688.$$
$$\Rightarrow NSG(T, Humidity) = 0.79 - 0.688 = 0.102.$$
$$\Rightarrow NSGr(T, Humidity) = \frac{NSG(T, Humidity)}{SplitInfo(T, Humidity)} = \frac{0.102}{1} = 0.102.$$

Similarly, we obtain:

$$NSGr(T, Outlook) = 0.055,$$
$$NSGr(T, Temp) = 0.027,$$
$$NSGr(T, Wind) = 0.017.$$

Hence, the attribute that will be assigned to the root node will be "Humidity" since it has the highest non-specificity gain ratio among all the attributes.

We get the NS-PDT tree as in Fig. 1.

For the training subsets $T^{\text{Humidity}}_{\text{high}}$ and $T^{\text{Humidity}}_{\text{normal}}$, we apply the same process as we did for the training set T until one of the stopping criteria holds.

The final NS-PDT tree induced by our algorithm is given by Fig. 2.

4.3. Classification procedure

Once the NS-PDT is constructed, we can classify any object having only values of all its attributes (see Section 2). As mentioned above, each leaf of our decision tree will be labeled by a possibility distribution over

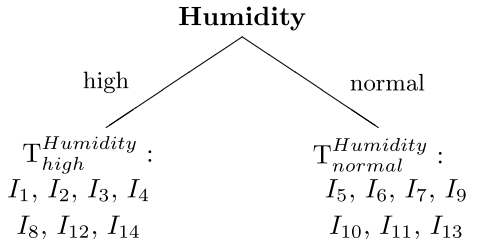

Fig. 1. First generated NS-PDT tree.

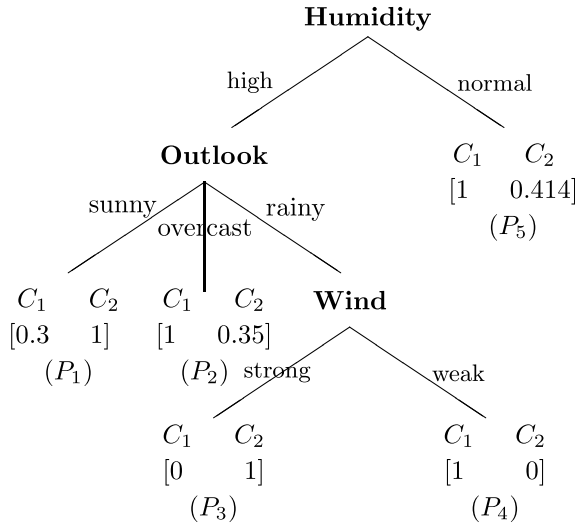

Fig. 2. Final NS-PDT tree.

the different class values. Hence, to make a decision about the class of a given object, the decision maker can take the fully possible class label (i.e. the class having a possibility degree equal to 1). Moreover, in cases where there may be unequal predefined costs depending on several classes in classification, the decision maker could opt for a cost-sensitive classification.

Example 2. Let us use the induced NS-PDT tree given in Example 1 to classify the object O_1 (Outlook:overcast, Temp:hot, Humidity: high, Wind: strong). Beginning from the root node of the tree, and following the edges relative to O_1's attribute values, we reach the leaf of the path (P_2) labeled by the possibility distribution $[1, 0.35]$. If we decide to take the fully possible class, object O_1 will be assigned the class C_1.

5. Non-specificity based possibilistic option decision trees (NS-PODT)

In [27], we have developed a variant of decision trees called *possibilistic option decision trees* (PODT) where each decision node can be split according to more than one attribute. Different options in the PODT are quantified via possibility distributions.

This section extends the NS-PDT approach, presented in the previous section, to deal with the uncertainty relative to the attribute selection step when several attributes appear as good discriminators. This extension have led to the so-called NS-PODT approach. We will, first, briefly recall basic parameters of the PODT approach. For more details, see [27].

5.1. Possibilistic option decision tree approach (PODT)

5.1.1. Attribute selection

As it is described, the standard building procedure [43] chooses at each decision node the attribute having the maximum or the minimum value (according to the context) of this measure, assuming that it leads to the smallest tree, and the remaining attributes are rejected.

For instance, suppose that at a node n, we find that $\mathrm{Gr}(T, A_1) = 0.87$ and $\mathrm{Gr}(T, A_2) = 0.86$. In standard decision tree building procedure, the node n will be split according to the values of A_1 whereas A_2 is rejected in spite of the fact that the two values are almost equal (the case of equality may also occur). Because of the heuristic aspect of the attribute selection measure as well as its one step lookahead nature, the situation is somewhat problematic: we agree that A_1 is a good splitting attribute at this level but nothing guarantees that A_1 is the best choice.

Hence, after computing the gain ratios of the different attributes, one should establish priorities between these candidate attributes according to the obtained values and select attributes that appears possible to a certain extent as well instead of choosing only the one with the highest gain ratio and rejecting all the remainders. Thus, the idea of the so-called possibilistic option decision tree approach (PODT) is to assign to each decision node n, a normalized possibility distribution π_{A_n} over the set of remaining attributes at this node, based on the set of gain ratios of the different attributes $GR = \{Gr(T_n, A_k) \text{ s.t. } A_k \in A_n\}$. T_n denotes the training subset relative to the node n.

Let A_n be the set of remaining attributes at a decision node n and GR the set corresponding to their gain ratios. We define a quantitative possibility distribution π_{A_n} by the following equation:

$$\pi_{A_n}(A_k) = \begin{cases} 0 & \text{if } Gr(A_k) \leqslant 0, \\ 1 & \text{if } Gr(A_k) = \max(GR), \\ \frac{Gr(A_k)}{Gr(A_k^*)} & \text{otherwise.} \end{cases} \tag{18}$$

We interpret $\pi_{A_n}(A_k)$ as the possibility degree that a given attribute A_k is reliable for the node n. An alternative manner to quantify the attributes was proposed by Hüllermeier in [23], but the characteristics of our possibility distribution is that it proportionally preserves the gap between the different attributes according to their gain ratios and it does not use any additional parameter. Once possibility degrees are generated for each attri-

bute, we use the *option technique* [6, 31], i.e., a decision node n will not be only split according to the best attribute A_k^* ($A_k^* = \arg\max_{A_k \in A_n}\{Gr(A_k)\}$) but rather for all attributes in the set A_n^* which we define by

$$A_n^* = \{A_k \in A_n \text{ s.t. } distance(A_k^*, A_k) \leqslant \Delta\}, \tag{19}$$

where $distance(A_k^*, A_k) = \pi_{A_n}(A_k^*) - \pi_{A_n}(A_k)$, A_n denotes the set of candidate attributes at the node n and Δ represents an arbitrary threshold varying in the interval $[0,1]$. The fixed value of Δ has a direct effect on the size of the tree. In fact, for a large (resp. small) value of Δ, the number of the selected attributes, at each node, will increase (resp. decrease) and hence, the tree will have a larger (resp. smaller) size. Note that when $\Delta = 0$, in some cases (i.e. when there is no equality between attributes' gain ratios) we recover a standard decision tree as C4.5 of Quinlan.

5.1.2. Partitioning strategy and stopping criteria

Since we can have more than one attribute at a given decision node n (an option node), the partitioning is realized as follows: For each attribute $A_k \in A_n^*$ and each value $v \in D(A_k)$, one outgoing edge is added to n. This edge is labeled with the value v and the possibility degree $\pi_{A_n}(A_k)$ which is interpreted as the reliability degree of that edge. Obviously, we keep the same stopping criteria as in standard decision trees.

Example 3. Let us use the original golf data set [37] to illustrate the induction of a possibilistic option decision tree (PODT). Let T' be the original training set labeled by the original crisp classes. The training set T' is given by Table 2:

Assume $\Delta = 0.4$ in Eq. (19).

Let us compute the gain ratios of the different attributes at the root node $n = 0$:

$$Gr(T'_0, Outlook) = \frac{\text{Gain}(T'_0, Outlook)}{SplitInfo(T_0, Outlook)} = \frac{0.246}{1.577} = 0.156;$$

$$Gr(T'_0, Temp) = \frac{Gain(T'_0, Temp)}{SplitInfo(T'_0, Temp)} = \frac{0.029}{1.556} = 0.018;$$

$$Gr(T'_0, Humidity) = \frac{\text{Gain}(T'_0, Humidity)}{SplitInfo(T'_0, Humidity)} = \frac{0.151}{1} = 0.151;$$

$$\text{Gr}(T'_0, Wind) = \frac{\text{Gain}(T'_0, Wind)}{SplitInfo(T'_0, Wind)} = \frac{0.048}{0.985} = 0.048.$$

We remark that the attribute "Outlook" has the highest gain ratio. Let us now, compute the possibility degrees of the different attributes, using Eq. (18), in order to define the set A_0^*:

Table 2
Training set

Outlook	Temp	Humidity	Wind	Class
Sunny	Hot	High	Weak	C_2
Sunny	Hot	High	Strong	C_2
Overcast	Hot	High	Weak	C_1
Rain	Mild	High	Weak	C_1
Rain	Cool	Normal	Weak	C_1
Rain	Cool	Normal	Strong	C_2
Overcast	Cool	Normal	Strong	C_1
Sunny	Mild	High	Weak	C_2
Sunny	Cool	Normal	Weak	C_1
Rain	Mild	Normal	Weak	C_1
Sunny	Mild	Normal	Strong	C_1
Overcast	Mild	High	Strong	C_1
Overcast	Hot	Normal	Weak	C_1
Rain	Mild	High	Strong	C_2

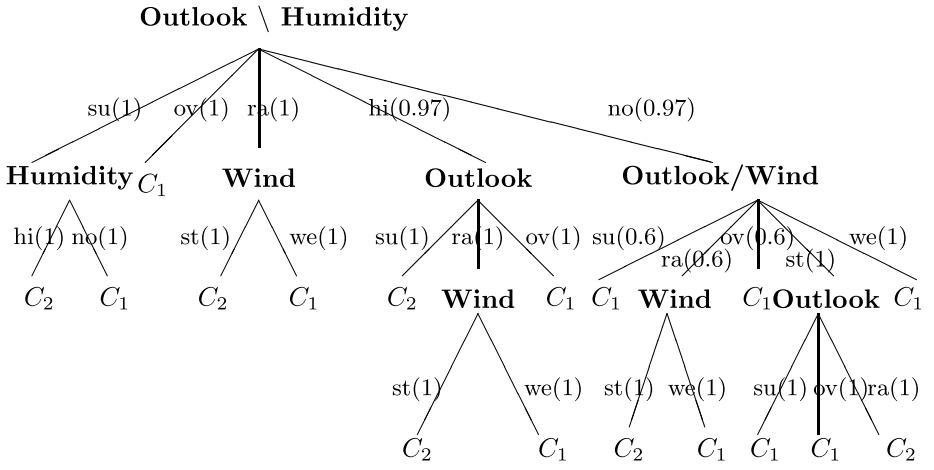

Fig. 3. Final possibilistic option tree.

$$\pi_{A_0}(Outlook) = 1$$
$$\pi_{A_0}(Temp) = \frac{Gr(T'_0, Temp)}{Gr(T'_0, Outlook)} = \frac{0.018}{0.156} = 0.12;$$
$$\pi_{A_0}(Humidity) = \frac{Gr(T'_0, Humidity)}{Gr(T'_0, Outlook)} = \frac{0.151}{0.156} = 0.97;$$
$$\pi_{A_0}(Wind) = \frac{Gr(T'_0, Wind)}{Gr(T'_0, Outlook)} = \frac{0.048}{0.156} = 0.31.$$

Given $\Delta = 0.4$, the set of attributes which will be assigned to the root n_0 of the possibilistic option tree is given by: $A_0^* = \{Outlook, Humidity\}$.

The possibilistic option tree induced from the training set T' ($\Delta = 0.4$ in Eq. (19)), which we denote by $PODT_{0.4}$, is given by Fig. 3. For clarity reasons, abbreviations of the attribute values are used instead of complete words (e.g. "ho" for the value "hot", "hi" for "high", "we" for "weak", etc.).

5.2. Building procedure in NS-PODT

Building a NS-PODT represents an extension of the building procedure of a NS-PDT to make this latter able to deal with the uncertainty relative to the attribute selection. As a consequence, we should modify some parameters of the NS-PDT approach, mainly, the attribute selection and the partitioning strategies.

In fact, instead of selecting only one attribute: the attribute maximizing *NSGr*, a set of possibly reliable attributes, i.e, attributes with high and close non-specificity gain ratios, will be assigned to the decision node at hand. We will follow the same procedure described in Section 5.1 but with the difference of using the *NSGr* attribute selection measure (Eq. (16)) instead of the traditional *gain ratio* (Eq. (4)) in Eq. (18).

$$\pi_{A_n}(A_k) = \begin{cases} 0 & \text{if } NSGr(A_k) \leqslant 0, \\ 1 & \text{if } NSGr(A_k) = \max(NSGR), \\ \frac{NSGr(A_k)}{NSGr(A_k^*)} & \text{otherwise,} \end{cases} \tag{20}$$

where $NSGR = \{NSGr(T_n, A_k) \text{ s.t. } A_k \in A_n\}$. We keep the same stopping criteria of the NS-PDT approach. Since more than one attribute can be selected in a decision node, the partitioning strategy of the NS-PODT will be the same as with the PODT approach (described in Section 5.1.2).

Let T be a training set composed of p objects $I_{j:1,\ldots,p}$ characterized by m discrete attributes $(A_1, A_2, \ldots, A_m)$ and belonging to the set of q mutually exclusive classes $C = \{C_1, C_2, \ldots, C_q\}$.

The NS-PODT algorithm uses a top down induction of decision tree approach. The different steps of the NS-PODT algorithm are described as follows:

Algorithm 1. NS-PODT building algorithm

Begin

1. Generate the root node of the non-specificity based possibilistic option tree including all the objects of the training set T.
2. Set the value of the threshold Δ (Eq. (19)) to control the number of attributes to be selected at each decision node: $|A_n^*|$, and hence, to control the size of the tree.
3. Verify **if** the generated node satisfies or not at least one of the stopping criteria listed in Section (4.2.2):
   (a) **If** yes, declare it as a leaf labeled by the appropriate possibility distribution.
   (b) **Else**, compute **for each** attribute, among those that have not been used so far, its non-specificity gain ratio (NSGr), then, generate its possibility degree of being a'reliable' splitting attribute using Eq. (20). Finally, choose the attributes satisfying Eq. (19) which will correspond to the root node of the NS-PODT tree relative to the whole training set.
4. Develop, **for each** value of each attribute in the set A_n^*, one outgoing edge marked with that value v and the possibility degree $\pi_{A_n}(A_k)$ of the corresponding attribute. The partitioning strategy (see Section 5.1.2) leads to several training subsets.
5. Create a root node relative to each induced training subset.
6. **Repeat** the same process for each training subset from **step 3**.
7. Stop when all the generated nodes of the latter level are declared as leaves.

End.

5.3. Classification procedure in NS-PODT

In addition to the classification of new certain instances (with unknown class values), i.e., ordinary instances, attribute values of which are known with certainty, the NS-PODT approach deals with the classification of uncertain instances, i.e., instances characterized by uncertain attribute values. In this section, we propose a method which ensures the classification of such instances. Uncertainty here is also handled in the possibilistic framework.

Given the set of attributes A, the instance to classify is described by a vector of possibility distributions $\overrightarrow{i} = (\pi_{A'_1}, \ldots, \pi_{A'_n})$. An attribute A_k whose value is known with *certainty* has exactly one value $v \in D(A_k)$, such that $\pi_{A'_k}(v) = 1$, and for all other values $v' \in D(A_k) - \{v\}$, $\pi_{A'_k}(v') = 0$. An attribute A_k whose value is *missing* is represented by a uniform possibility distribution, i.e., $\forall v \in D(A_k), \pi_{A'_k}(v) = 1$. Table 3 gives an example of an uncertain instance $\overrightarrow{i_1}$ to classify.

In order to classify an uncertain instance (e.g. $\overrightarrow{i_1}$) within a NS-PODT tree, we need to carry out the following steps:

Step one: Instance propagation

At each option node of a NS-PODT tree, the instance to classify can branch in different directions depending on the chosen attribute to test on. To each one of these attributes, we have assigned a possibility degree $\pi_{A_n}(A_k)$ (Eq. (20)) indicating the possibility that a given attribute is reliable for a given option node n.

Table 3
Instance $\overrightarrow{i_1}$

π'_{outlook}	π'_{temp}	π'_{humidity}	π'_{wind}
Sunny 1	Hot 1	High 1	Strong 0
Overcast 0.5	Mild 1	Normal 0.5	Weak 1
Rain 0.7	Cool 0.4		

Thus, throughout a given NS-PODT, whenever an instance follows an attribute A_k, the related possibility distribution in the instance to classify ($\pi_{A'_k}$) should be discounted according to the possibility degree of the followed attribute ($\pi_{A_n}(A_k)$) using Eq. (9). The resulting discounted possibility degrees will replace the degrees labeling the NS-PODT.

Step two: Exploring paths

Once the propagation is made within the NS-PODT tree (step 1), we should explore all its paths in order to determine their corresponding possibility degrees based on the 'new' discounted possibility degrees labeling the tree. Since a path represents a conjunction of edges, we have used the *minimum* conjunctive operator to define the possibility degree of a path $p = (n_0, \ldots, n_k)$ as

$$\pi_{\text{path}}(p) = \min_{0 \leqslant i < l(p)-1} \pi_{\text{edge}}((n_i, n_{i+1})), \tag{21}$$

where $\pi_{\text{edge}}((n_i, n_{i+1}))$ denotes the possibility degree labeling the edge (n_i, n_{i+1}) and $l(p)$ denotes the length (number of nodes) of the path p. We interpret $\pi_{\text{path}}(p)$ as the reliability degree of the path p. It indicates to what extent the path is supporting the instance to classify according to its attribute values (i.e. the possibility degrees labeling its attributes).

Step three: Exploring leaves

Recall that each leaf in a NS-PODT tree is labeled by a possibility distribution on the different class values. Therefore, one should refine the possibility distribution labeling each leaf by the reliability degree of the path leading to that leaf by applying the discounting formula given by Eq. (9). Obviously, possibility distributions labeling leaves of fully reliable paths of the tree (i.e. with reliability degree equal to 1), will remain unchanged. In the case of fully unreliable paths, i.e., paths which are not supporting the object to classify, all possibility distributions will move to the total ignorance in order to be ignored later.

Once discounted, the idea is to rank, by level of non-specificity, the resulted possibility distributions according to their non-specificities, i.e., from the least non-specific (the most specific) to the most non-specific (the least specific). Obviously, each level may contain more than one possibility distribution. Having the different sets of possibility distributions, ranked in decreasing order by level of non-specificity, the classification result of the instance at hand is done according to the following algorithm:

Algorithm 2. NS-PODT classification algorithm

Begin

1. Start by the first level ($level_1$).
2. **If** the current level ($level_i$) contains *only one* possibility distribution **OR** *identical* possibility distributions: take one of these possibility distributions as the classification result.
3. **Else**, i.e., the current level contains *several different* possibility distributions:
   (a) Compute $\pi_{\text{Rep}}^{level_i}$; the representative possibility distribution of the set of possibility distributions present at level i.
   (b) **If** the non-specificity of $\pi_{\text{Rep}}^{level_i}$ is below or equal to the non-specificity of $\text{level}_{(i+1)}$: take $\pi\_Rep^{level_i}$ as the classification result.
   (c) **Else**: move to the next level and return to 2.

End.

Note that, by convention, $level_{(n+1)}$ has the maximum non-specificity value which is equal to $\log_2|C|$. Recall that C represents the set of possible classes of the problem. If $level_{(n+1)}$ is reached, the classification result π^{res} satisfies $\pi^{\text{res}} = \arg\min_{i=1}^{n}\{U(\pi_{\text{Rep}}^{level_i})\}$.

In the proposed procedure, we chose to take into account the non-specificity of the resulting possibility distributions in order to make our decision. The choice of the more specific possibility distribution is justified by the fact that the path leading to this latter, better supports the object to classify (according to the possibility degrees labeling object's attributes). Because we could have cases of equal non-specificities, we proposed to use levels of non-specificities. Each level gathers together possibility distributions with equal non-specificities. We

move to a next level only when its non-specificity is lower than the non-specificity of the representative possibility distribution of the current level.

Example 4. Suppose we have to classify the instance $\overrightarrow{i_1}$ given in Table 3 using the induced NS-PDT tree of Example 1 given in Section 4. Note that, in this case, the induced NS-PDT tree is equivalent to an NS-PODT_0: a NS-PODT_Δ tree with $\Delta = 0$ (no equality between attributes has been occurred in any node).

Step one: *Instance propagation*
Starting from the root node of the NS-PODT_0 (see Fig. 2), the instance $\overrightarrow{i_1}$ will follow all the edges of the tree. According to the reliability degree of each followed edge, we will discount the corresponding possibility distribution π'_{A_k} as mentioned above. Note that in the NS-PODT_0, all edges are fully reliable (reliability degrees equal to 1) since we have obtained only one attribute in each decision node. Consequently, the different edges of the NS-PODT_0 will be labeled by the discounted possibility distributions which, in this particular case, will remain unchanged.
Step two: *Exploring paths*
Let us compute the possibility degree relative to each path using Eq. (21):
(P_1): $\min(1,1) = 1$
(P_2): $\min(1,0.5) = 0.5$
(P_3): $\min(1,0.7,0) = 0$
(P_4): $\min(1,0.7,1) = 0.7$
(P_5): 0.5

Step three: *Exploring leaves*
The possibility distribution labeling each leaf of the tree will be discounted according to the possibility degree of the path leading to that leaf:
$(P_1)^1$: [0.3 1] ⇒[0.3 1]
$(P_2)^{0.5}$: [1 0.35] ⇒[1 0.5]
$(P_3)^0$: [0 1] ⇒[1 1]
$(P_4)^{0.7}$: [1 0] ⇒[1 0.3]
$(P_5)^{0.5}$: [1 0.414] ⇒[1 0.5]

The rank is given by

$$level_1 : \{[0.31], [10.3]\}$$
$$level_2 : \{[10.5], [10.5]\}$$
$$level_3 : \{[11]\}$$

⇒According to the procedure proposed above, the classification result will be the distribution: [1 0.5] which means that the class of $\overrightarrow{i_1}$ is C_1 with a possibility degree equal to 1 and C_2 with a possibility degree of 0.5.

6. Experimental results

In a first step, the main purpose of our experimental study was to show that exploiting uncertain data for decision tree induction by using the proposed NS-PDT approach is usually better than the obvious alternative, namely to ignore such data and learn with a standard decision tree algorithm from the remaining (exactly labeled) examples. Afterwards, the experiments will show that the extended version, namely, NS-PODT outperforms NS-PDT in terms of classification performance, hence, paying off the increased complexity of the former.

For the evaluation of different possibilistic decision tree approaches proposed in this paper, we have developed a possibilistic decision tree toolbox (PD2T) implemented with Matlab 6.5. The toolbox includes a home-made implementation of the C4.5 algorithm, an implementation which is faithful to the original C4.5 as presented by Quinlan [43]. From this implementation, we have derived all of the PODT, the NS-PDT and

Table 4
Description of databases

Database	# Data	# Attributes	# Classes
Wisconsin breast cancer	699	8	2
Voting	497	16	2
Solar flare	1389	10	3
Balance scale	625	4	3
Nursery	12960	8	5

the NS-PODT approaches. We have also used the well-known Weka data mining tool [50], especially, the J48 classifier (an implementation of C4.5).

The experimental study is based on several data sets selected from the U.C.I. repository of machine learning databases [37]. A brief description of these data sets is given in Table 4. #Data, #attributes, #classes denote respectively the total number of instances, the number of attributes and the number of classes.

In order to conduct our experiments for both NS-PDT and NS-PODT approaches, we have "contaminated" these data sets by transforming the original crisp classes by possibility distributions over the different classes. We used levels of uncertainty ($L\%$) when generating these possibilistic training sets. More precisely, for each training instance from the $L\%$ randomly chosen instances, we have assigned a possibility degree equal to 1 to the original class and a random possibility degree (from $[0,1[$) to the remainders in an uniform way. To each one of the remaining $(100 - L)\%$ instances of the original training set, we have assigned a completely sure possibility distribution corresponding to the original crisp instance's class. For our experiments, we have varied L from 0 (crisp training set) to 50 (half of the training instances has an uncertain class label).

In order to determine the accuracy of the induced trees, we have used two criteria, the first is relative to the percentage of correct classification (PCC) expressed by

$$PCC = \frac{\text{number of well classified instances}}{\text{total number of classified instances}} \times 100 \tag{22}$$

and the second corresponds to a distance based criterion (PCC_dist) which we propose as a new criterion that is more appropriate to the possibilistic context:

$$PCC\_dist = \frac{\sum_{\overrightarrow{I_j} \in \text{classified instances}} D(\overrightarrow{I_j})}{\text{total number of classified instances}} \times 100, \tag{23}$$

where $D(\overrightarrow{I_j}) = 1 - \frac{d(\overrightarrow{I_j})}{|C|}$ and $d(\overrightarrow{I_j}) = \sum_{i=1}^{|C|} (\pi^{\text{res}}(C_i) - \pi^j(C_i))^2$.

Recall that within our possibilistic decision tree approach, the classification result is given in the form of a possibility distribution (π^{res}). Thus, the idea is to choose for each instance to classify the class having the highest possibility degree (equal to 1). If more than one class is obtained, then one of them is chosen randomly. The obtained class is considered as the class of the testing instance. Consequently, *number of well-classified instances* in Eq. (22) corresponds to the number of testing instances for which the class obtained by the possibilistic decision tree approach (the more plausible class) is the same as their real more plausible class.

The limitation of the PCC criterion, in our case, is that it chooses randomly one of the more plausible classes which may miss-classify some instances. Moreover, even when there is only one more plausible class, focusing on that class and ignoring the rest of the classes (classes with possibility degrees different to 1) is problematic. In fact, ignoring the rest of the degrees implies ignoring a part of the information given by the resulting possibility distribution (π^{res}).

So, we were inspired by the criterion proposed in [2] to define the PCC_dist criterion which takes into account the mean distance relative to all the classified testing instances: the average of the distances between the resulting possibility distribution (π^{res}) and the real possibility distribution (π^j) of each classified instance $\overrightarrow{I_j}$. When $D(\overrightarrow{I_j})$ is close to 100, the classifier is good whereas when it falls to 0, it is considered as a bad classifier.

The experimental methodology is as follows: for each training set and for each uncertainty level L (from 0% to 50%), we have induced a NS-PDT tree. On the other hand, a C4.5 tree was induced from the corresponding training set, i.e., the ordinary training set from which we have discarded the $L\%$ instances to which we have assigned imprecise class labels since the C4.5 algorithm can not deal with such instances.

Then, both approaches are evaluated on the same testing sets: ordinary testing sets for C4.5 trees have been used and their corresponding testing sets (with completely sure possibility distributions on the original class labels) for NS-PDT trees: this corresponds to one iteration of the 10-fold cross validation process used for the evaluation of the approach.

Table 5 reports the different obtained results after varying the training sets' level of uncertainty $L\%$ from 0% to 50% for each database. *MPCC* denotes the mean *PCC* (complemented by standard deviation) of the induced decision trees for NS-PDT, C4.5-U (Unpruned) and C4.5-P (Pruned) approaches after a 10-fold cross validation testing process.

Table 5 shows that the NS-PDT approach gives interesting results when compared with the famous C4.5. Note that the aim of this comparison is not to directly compare the two approaches since C4.5 is only used in certain environments while the NS-PDT approach deals with both certain and uncertain environments.

From Table 5, we can see that for low levels of uncertainty and especially, when there is no uncertainty in the training set ($L = 0\%$), accuracy rates of the state-of-the-art C4.5 approach (regardless of pruning) are slightly higher than those of NS-PDT. Interestingly, the NS-PDT approach begins to perform better than C4.5 for higher levels of uncertainty (e.g. $L = 40\%$ for W.B.Cancer dataset, ($L = 20\%$, Voting), ($L = 30\%$, Solar Flare) and ($L = 10\%$, Balance and Nursery)). In fact, classification accuracies of both approaches decrease when the level of uncertainty increases. This can be explained by the fact that the higher the level of uncertainty ($L\%$), the less informative the training set becomes (consequently, the harder the learning becomes), and therefore the less accurate the predictions are. In spite of this decrease in accuracy, we can see that the classification rate of NS-PDT is greater than the one of C4.5 (even for the C4.5-P) and this difference becomes more important for training sets with higher proportions of uncertain instances.

Besides, from the above reported percentages, we can see that, for the Balance (three classes) and Nursery (five classes) data sets, the NS-PDT approach rapidly outperforms C4.5 (only from $L = 10\%$). This represents an interesting result. Indeed, it shows that the NS-PDT approach is well suited for problems with large number of classes where uncertainty becomes more relevant and more difficult to manage.

The principal result of this table is that, generally, rejecting training instances, classes of which are imprecisely defined, is not a good practice and reduces the accuracy of the induced classifier "(and especially for muti-class problems)". This issue can be avoided and well handled by the use of the proposed NS-PDT approach which can exploit the information contained in imprecise labels.

Regarding the complexity, let us note that for each uncertainty level $L\%$, the obtained NS-PDT trees and C4.5-U trees have almost the same number of leaves (with a slight increase for the NS-PDT approach). This can be due to the fact that C4.5-U trees are induced from smaller training sets, i.e., training sets from which we have discarded imprecisely labeled instances.

Table 5
Results for C4.5 and NS-PDT (MPCC and standard deviation)

Database	Method	$L = 0\%$	$L = 10\%$	$L = 20\%$	$L = 30\%$	$L = 40\%$	$L = 50\%$
W.B. cancer	C4.5-U	94.54(1.1)	93.86(1.4)	91.63(2.3)	91.05(2.5)	90.49(2.8)	90.11(3.2)
	C4.5-P	95.26(1.2)	94.45(1.4)	93.10(2.1)	92.63(2.5)	92.11(2.6)	90.23(3.2)
	NS-PDT	93.94(1.2)	93.85(1.2)	93.16(1.7)	92.32(2.2)	92.13(2.3)	91.78(2.5)
Voting	C4.5-U	94.56(3.2)	93.42(3.2)	92.23(3.5)	90.15(3.8)	89.59(4.3)	87.27(4.6)
	C4.5-P	95.57(3.0)	94.12(3.2)	92.58(3.4)	91.74(3.5)	90.31(3.7)	88.59(4.1)
	NS-PDT	93.86(3.0)	93.15(3.1)	92.76(3.3)	92.43(3.4)	90.82(3.7)	89.88(3.9)
Solar flare	C4.5-U	81.96(3.3)	80.38(3.5)	78.68(3.5)	77.03(3.7)	76.67(3.7)	74.37(3.9)
	C4.5-P	82.42(3.1)	81.14(3.0)	80.56(3.3)	78.27(3.3)	76.89(3.7)	75.17(3.7)
	NS-PDT	81.37(3.0)	80.79(3.1)	80.21(3.3)	79.63(3.4)	78.55(3.4)	76.38(3.6)
Balance	C4.5-U	78.48(4.2)	77.12(4.3)	75.39(4.7)	74.78(5.3)	72.42(5.6)	70.38(5.7)
	C4.5-P	74.48(4.0)	74.12(4.2)	73.77(4.4)	73.19(4.9)	72.32(5.2)	69.81(5.3)
	NS-PDT	78.18(4.2)	77.86(4.2)	76.12(4.3)	75.63(4.7)	75.32(4.8)	74.84(5.0)
Nursery	C4.5-U	98.78(0.8)	96.38(1.3)	95.27(1.4)	94.45(1.6)	93.73(2.3)	92.81(2.6)
	C4.5-P	97.05(1.1)	94.22(1.1)	91.73(1.2)	91.13(1.5)	90.42(2.3)	90.13(2.3)
	NS-PDT	97.72(1.1)	97.42(1.1)	97.21(1.3)	97.03(1.3)	96.51(1.4)	95.27(2.3)

Table 6
NS-PDT: Mean *PCC_dist* and standard deviation

$L\%$	0%	10%	20%	30%	40%	50%
W.B. cancer	95.56(0.8)	95.23(0.9)	94.56(0.9)	94.11(1.1)	93.28(1.5)	93.17(1.5)
Voting	95.57(2.7)	95.12(2.7)	94.73(2.9)	93.65(3.1)	92.88(3.3)	92.25(3.3)
Solar flare	84.27(1.4)	83.68(1.6)	82.54(1.6)	82.12(1.7)	81.44(2.0)	80.79(2.3)
Balance scale	82.83(0.6)	82.42(0.7)	81.93(0.9)	80.75(1.1)	80.18(1.1)	78.71(1.4)
Nursery	98.34(0.9)	97.75(1.2)	96.97(1.3)	96.51(1.3)	95.69(1.4)	95.26(1.6)

Table 6 reports the mean *PCC_dist* value (complemented by standard deviation) for each database for the different levels of uncertainty after a 10-fold cross validation process. Note that high values of the *PCC_dist* criterion do not only imply that the induced trees are accurate but also imply that the possibility distributions provided by the induced NS-PDT trees are of high quality and faithful to the original possibility distributions. Again, from Table 6, we can see that *PCC_dist* values decrease when $L\%$ increases (for the same explanation provided above for Table 5).

Let us move to Table 7 for the evaluation of the extension of NS-PDT, namely the NS-PODT approach to show the effect of the "optional" splitting on the accuracy of the induced trees.

We followed the same experimental strategy: for each value of Δ (from 0 to 0.5), we varied the level of uncertainty $L\%$ (from 0 to 50). Note that the variation of Δ differs from one data set to another. In fact, this is done experimentally: since for larger values of Δ the tree size becomes more and more important, then, we stopped the variation of Δ when we reach a compromise between the size and the accuracy of the induced tree, i.e., a reasonable size and a relatively high accuracy (compared with the initial tree NS-PODT_0).

Table 7 reports, for each data set, the mean accuracy rate (MPCC) of the induced NS-PODT_Δ trees after running a 10-fold cross validation testing process. In our experiments, we have varied Δ from 0 to 0.5 but we only report results for the special case ($\Delta = 0$) and for the optimal value of Δ (marked with $*$) which induces the most accurate NS-PODT tree. Note that in [27], we have shown that the performance of PODT is neither a monotone increasing nor a monotone decreasing function of the parameter Δ. We obtained similar results with NS-PODT.

From Table 7, we notice that for some data sets (voting, solar flare and nursery), NS-PODT_0 is equivalent to NS-PDT which means that there is no equality between the discriminative power of the attributes during the whole building process of the induced trees. Besides, the value of the optimal Δ is purely experimental and depends on the used data set ($\Delta = 0.1$ for W.B.cancer and balance data, 0.4 for solar flare and 0.2 for nursery).

Table 7
Results for NS-PODT_Δ (MPCC and standard deviation)

Database	Method	Δ	$L = 0\%$	$L = 30\%$	$L = 50\%$
W.B. cancer	NS-PDT	–	93.94(1.2)	92.32(2.2)	91.78(2.5)
	NS-PODT	0	94.75(1.1)	93.16(2.1)	92.27(2.5)
	NS-PODT	0.1*	95.33(1.2)	93.81(2.2)	93.29(2.4)
Voting	NS-PDT	–	93.86(3.0)	92.43(3.4)	89.88(3.9)
	NS-PODT	0	93.86(3.0)	92.43(3.4)	89.88(3.9)
	NS-PODT	0*	93.86(3.0)	92.43(3.4)	89.88(3.9)
Solar flare	NS-PDT	–	81.37(3.0)	79.63(3.4)	76.38(3.6)
	NS-PODT	0	81.37(3.0)	79.63(3.4)	76.38(3.6)
	NS-PODT	0.4*	83.85(2.7)	82.47(3.2)	80.25(3.4)
Balance	NS-PDT	–	78.18(4.2)	75.63(4.7)	74.84(5.0)
	NS-PODT	0	79.22(3.9)	77.38(4.6)	76.45(4.8)
	NS-PODT	0.1*	79.46(3.8)	77.53(4.5)	76.82(4.8)
Nursery	NS-PDT	–	97.72(1.1)	97.03(1.3)	95.27(2.3)
	NS-PODT	0	97.72(1.1)	97.03(1.3)	95.27(2.3)
	NS-PODT	0.2*	98.57(0.9)	97.83(1.1)	96.31(2.2)

The results given in Table 7 confirm the results reported in our previous work [27]. Indeed, even with the new approach which uses new parameters, considering more than one attribute in some decision nodes, i.e, those that appear as possibly good discriminators instead of rejecting them may increase the classification accuracy of the resulted trees. In fact, except the Voting data set, we can see that for each data set, the NS-PODT approach outperforms NS-PDT in terms of classification performance, so, paying off the increased complexity of NS-PODT. Hence, we can conclude that using the "optional" splitting when building decision trees might enhance the accuracy of the induced trees.

7. Related works

Many decision tree approaches under uncertainty were proposed in the literature, namely, probabilistic decision trees [7,42], belief decision trees [10,16,46,47], possibilistic decision trees [2,4,23,27], fuzzy decision trees [26,34,38–40,52] and credal decision trees [1]. The difference between the existing approaches lies mainly in the type of uncertainty presented to the problem at hand (incompleteness, conflict, imprecision, vagueness, etc.) and especially in the way of dealing with that uncertainty when building the tree.

Within the possibilistic framework, Borgelt et al. [4] proposed a possibility based attribute selection measure which they used for the induction of possibilistic decision trees. In their work, the authors take the probability distributions, i.e. the frequency distributions of the instances reaching each node as possibility distributions (an interpretation which is based on the context model of possibility theory [19, 32]). As the role non-specificity plays in possibility theory is similar to that of Shannon entropy in probability theory, a non-specificity based attribute selection measure is constructed from it in the same way as the information gain ratio criterion of the well-established C4.5 algorithm [43] is constructed from Shannon entropy. This measure is slight different from the measure we used in this paper but it is the way of using this measure that is totally different. In fact, in our approach, the parameters of the induction method and the induction method itself are different from Borgelt's work which keeps the same parameters of the C4.5 algorithm. Furthermore, in our work, the need of using a non-specificity based attribute selection measure suggests itself since instances' classes in the training set are given by normalized possibility distributions. On the contrary, in Borgelt's work, instances' classes in the training set are crisp.

Still within the possibilistic framework, we can mention two additional works. First, the work proposed by Hüllermeier [23] in which he used a possibilistic branching within the lazy decision tree technique. In this work, the author has not dealt with any uncertainty in the training set (building phase) nor in the classification phase. In the second work [2], the authors have only dealt with uncertainty in the classification phase. More precisely, they extended ordinary decision trees by proposing a method for the classification of instances having uncertain attribute values (given by qualitative possibility distributions) using the leximin–leximax criterion [36].

As we have proposed a non-specificity based attribute selection measure in our approach, it is important to note that the idea of using non-specificity for building decision trees is not new. In fact, it has been used with other uncertainty theory frameworks. In 1995, Yuan et al. proposed a measure of *classification ambiguity* which is defined from both a measure of fuzzy subsethood and a measure of non-specificity. This measure was used as an attribute selection measure to construct *fuzzy decision trees* [52]. Since then, many other fuzzy decision tree techniques were proposed [26,34,38–40]. In these different works, fuzzy set theory is used either to manage fuzzy attributes and/or fuzzy labels in the training set or to search for the degree of softness in every node of the built tree or to integrate some interesting fuzzy tools during the building phase.

More recently, Denoeux et al. have dealt with the induction of belief decision trees from data with partially defined classes presented in the form of basic belief assignments (b.b.a.) [10]. The authors have used a total uncertainty criterion as a measure of discrimination based on both measures of non-specificity and conflict relative to belief function theory. In the same context, another approach was proposed by Elouedi et al. for inducing belief decision trees (BDT) [14,15] and pruning methods for this approach have been recently proposed [46]. In the BDT approach, the authors have presented two attribute selection measures using the belief function formalism, one parallel to Quinlan's measure based on Shannon entropy (the averaging approach), the other close in spirit to the transferable belief model [45] (the conjunctive approach). Note that, differently to the approaches listed above, this approach did not use any non-specificity measure. The common point

between belief decision trees and our approach is that we start with the "same" hypothesis: uncertain classes in the training set, but it is the theory and the interpretation of the two theories that differ.

Another decision tree approach using non-specificity is the one proposed by Abellan et al. [1]. In their approach, which has no relation with the ours, the authors used the imprecise Dirichlet model [48] to estimate the probabilities of instances' classes reaching a given node. More specifically, the probability of each class is transformed into a probability interval (imprecise probability) to obtain what they called a credal set (a convex set of probability distributions). Then, they proposed a total uncertainty measure (non-specificity + entropy) to assess the impurity of the different nodes (credal sets) of the tree under construction.

Several non-standard classification techniques were proposed in the literature. We can mention the belief k-nearest neighbor classifier [8] and the belief neural network classifiers [9] both based on Dempster–Shafer theory. We can also mention the possibilistic instance based learning approach [24]. In [25], the author reviewed some typical applications of fuzzy set theory to machine learning techniques. More recently, a fuzzy lattice reasoning classifier [28] was proposed to induce rules in a mathematical lattice data domain. This classifier has the advantage of being incremental and able to deal with missing data.

8. Conclusion

In this paper, we have presented a classification technique under an uncertain environment. First, we have proposed the NS-PDT approach, then we have extended it to obtain the NS-PODT. It is the result of the combination between the decision tree technique and possibility theory. This combination makes the former able to deal with uncertainty which can appear in different parameters of the classification problem. Obviously, we are interested in the uncertainty that has a possibilistic nature.

In a first part, we considered the case where instances's classes in the training set are given by possibility distributions. As a consequence, when building the tree, instances reaching the different nodes of the tree will be characterized by possibility distributions over the different classes of the problem instead of crisp classes. In order to adapt the ordinary decision tree learning algorithm to such a situation, we have proposed a non-specificity based attribute selection measure (the NSGr criterion) instead of Quinlan's *gain ratio* criterion which is based on the probabilistic Shannon entropy. Then, we have defined the different parameters of the so-called NS-PDT approach.

In a second part, based on our previous work [27], we have extended the NS-PDT approach to deal with another kind of uncertainty which is hidden in the decision tree building procedure. More precisely, we have made the NS-PDT approach able to deal with the uncertainty related to the choice of an attribute among a set of attributes with equally or nearly equal NSGr values. Hence instead of selecting only one attribute in a given decision node and rejecting the others, with the so-called NS-PODT approach, we allow the selection of more than one attribute: the most possibly reliable ones. Obviously, each attribute will be characterized by its possibility degree. The different parameters of this approach have been also detailed.

Another interesting contribution lies in the use of the induced NS-PODT tree, i.e., in the classification (inference) task. In fact, in addition to the classification of instances having crisp attribute values, we have proposed a whole procedure allowing the classification of instances having uncertain attribute values, i.e., we considered the case where the knowledge about the value of some attributes is represented by a possibility distribution.

After the presentation of the theoretical concepts underlying the different proposed possibilistic decision tree approaches, we have evaluated them by applying them on commonly used data sets obtained from the U.C.I. repository [37]. In a first part, experimental studies have shown that ignoring training instances, classes of which are imprecisely defined, is not a good practice and reduces the accuracy of the induced classifier which can be avoided and well handled by the use of the proposed NS-PDT approach.

In a second part, the extension of the NS-PDT approach, namely the NS-PODT approach has shown that the classification accuracy can increase when varying Δ until reaching a specific value which is purely experimental. This value is relatively small and hence the time and space complexity remain reasonable. We plan to use our approach for the intrusion detection problem where the knowledge about connection types (normal, a specific attack type) is, by nature, afflicted with uncertainty. We also think that the pruning issue should be investigated and aim to extend our approach to handle continuous attributes in the future.

Acknowledgement

We would like to thank the editor and the anonymous reviewers for their constructive comments which have helped us to improve the paper considerably.

References

[1] J. Abellan, S. Moral, Upper entropy of credal sets. Applications to credal classification, International Journal of Approximate Reasoning 39 (2–3) (2005) 235–255.
[2] N. Ben Amor, S. Benferhat, Z. Elouedi, Qualitative classification and evaluation in possibilistic decision trees, in: Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'04), Budapest, Hungary, vol. 2, 2004, pp. 653–657.
[3] J.C. Bezdek, D. Dubois, H. Prade, Fuzzy Sets in Approximate Reasoning and Information Systems, The Handbooks of Fuzzy Sets Series, Kluwer Academic Publishers, 1999.
[4] C. Borgelt, J. Gebhardt, R. Kruse, Concepts for probabilistic and possibilistic induction of decision trees on real world data, in: Proceedings of the 4th European Congress on Fuzzy and Intelligent Technologies (EUFIT'96), Aachen, 1996, pp. 1556–1560.
[5] L. Breiman, J.H. Friedman, R.A. Olshen, C.J. Stone, Classification and Regression Trees, Wadsworth & Brooks, Monterey, CA, 1984.
[6] W. Buntine, Learning classification trees, in: D. Hand (Ed.), Artificial Intelligence Frontiers in Statistics, Chapman & Hall publishers, London, 1991, pp. 182–201.
[7] A. Ciampi, E. Diday, J. Lebbe, E. Périnel, R. Vignes, Growing a tree classifier with imprecise data, Pattern Recognition Letters 21 (2000) 787–803.
[8] T. Denoeux, A *k*-nearest neighbor classification rule based on Dempster–Shafer theory, IEEE Transactions on Systems, Man and Cybernetics 25 (05) (1995) 804–813.
[9] T. Denoeux, A neural network classifier based on Dempster–Shafer theory, IEEE Transactions on Systems, Man, and Cybernetics, Part A 30 (2) (2000) 131–150.
[10] T. Denoeux, M. Skarstein-Bjanger, Induction of decision trees from partially classified data, in: Proceedings of the 2000 IEEE International Conference on Systems, Man and Cybernetics (SMC'00), IEEE, Nashville, TN, 2000, pp. 2923–2928.
[11] T. Denoeux, L.M. Zouhal, Handling possibilistic labels in pattern classification using evidential reasoning, Fuzzy Sets and Systems 122 (3) (2001) 47–62.
[12] D. Dubois, H. Prade, Unfair coins and necessity measures: towards a possibilistic interpretation of histograms, Fuzzy Sets and Systems 10 (1) (1985) 15–20.
[13] D. Dubois, H. Prade, Possibility Theory: An Approach to Computerized Processing of Uncertainty, Plenum Press, New York, 1988.
[14] D. Dubois, H. Prade, Representation and combination of uncertainty with belief functions and possibility measures, Computational Intelligence 4 (3) (1988) 244–264.
[15] 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 in Knowledge-Based Systems (IPMU'2000), Madrid, Spain, 2000, pp. 141–148.
[16] Z. Elouedi, K. Mellouli, P. Smets, Belief decision trees: theoretical foundations, International Journal of Approximate Reasoning 28 (2001) 91–124.
[17] U.M. Fayyad, K.B. Irani, On the handling of continuous-valued attributes in decision tree generation, Machine Learning 8 (1992) 87–102.
[18] J.H. Friedman, R. Kohavi, Y. Yun, Lazy decision trees, in: Proceedings of the 13th National Conference on Artificial Intelligence, 1996, pp. 717–724.
[19] J. Gebhardt, R. Kruse, Learning possibilistic networks from data, in: Proceedings of the 5th International Workshop on Artificial Intelligence and Statistics, Fort Lauderdale, 1995, pp. 233–244.
[20] D. Harmanec, Measures of uncertainty and information. <http://www.sipta.org>, 1999.
[21] R.V. Hartley, Transmission of information, Bell System Technical Journal 7 (1928) 535–563.
[22] M. Higashi, G.J. Klir, Measures of uncertainty and information based on possibility distributions, International Journal of General Systems 9 (1) (1983) 43–58.
[23] E. Hüllermeier, Possibilistic induction in decision tree learning, in: Proceedings of the 13th European Conference on Machine Learning (ECML'02), Helsinki, Finland, 2002, pp. 173–184.
[24] E. Hüllermeier, Possibilistic instance-based learning, Artificial Intelligence 148 (1–2) (2003) 335–383.
[25] E. Hüllermeier, Fuzzy methods in machine learning and data mining: status and prospects, Fuzzy Sets and Systems 156 (3) (2005) 387–406.
[26] C.Z. Janikow, Fuzzy decision trees: issues and methods, IEEE Transactions on Systems, Man and Cybernetics—Part B: Cybernetics 28 (1) (1998) 1–14.
[27] I. Jenhani, Z. Elouedi, N. Ben Amor, K. Mellouli, Qualitative inference in possibilistic option decision trees, in: Proceedings of the 8th European Conference in Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU'05), Barcelona, Spain, 2005, pp. 944–955.
[28] V.G. Kaburlasos, I.N. Athanasiadis, P.A. Mitkas, Fuzzy lattice reasoning (FLR) classifier and its application for ambient ozone estimation, International Journal of Approximate Reasoning 45 (1) (2007) 152–188.

[29] G.J. Klir, T.A. Folger, Fuzzy Sets, Uncertainty, and Information, Prentice-Hall, Englewood Cliffs, NJ, 1988.
[30] G.J. Klir, M.J. Wierman, Uncertainty-Based Information: Elements of Generalized Information Theory, Studies in Fuzziness and Soft Computing, vol. 15, 1998.
[31] R. Kohavi, C. Kunz, Option decision trees with majority votes, in: Proceedings of the 14th International Conference on Machine Learning (ICML'97), Nashville, TN, USA, 1997, pp. 161–169.
[32] R. Kruse, J. Gebhardt, F. Klawonn, Foundations of Fuzzy Systems, John Wiley and Sons, Chichester, 1994.
[33] S. Kwan, F. Olken, D. Rotem, Uncertain, incomplete, and inconsistent data in scientific and statistical databases, in: Proceedings of the Workshop on Uncertainty Management in Information Systems: From Needs to Solutions, Mallorca, Spain, 1992, pp. 64–91.
[34] L.F. Mendonça, S.M. Vieira, J.M.C. Sousa, Decision tree search methods in fuzzy modeling and classification, International Journal of Approximate Reasoning 44 (2) (2007) 106–123.
[35] A. Motro, Sources of uncertainty, imprecision and inconsistency in information systems, in: Proceedings of the Workshop on Uncertainty Management in Information Systems: From Needs to Solutions, 1996, pp. 9–34.
[36] H. Moulin, Axioms for Cooperative Decision-making, Cambridge University Press, 1988.
[37] P.M. Murphy, D.W. Aha, UCI repository of machine learning databases, 1996. <http://mlearn.ics.uci.edu/MLRepository.html>.
[38] C. Olaru, L. Wehenkel, A complete fuzzy decision tree technique, Fuzzy Sets and Systems 138 (2003) 221–254.
[39] W. Pedrycz, Z.A. Sosnowski, The design of decision trees in the framework of granular data and their application to software quality models, Fuzzy Sets and Systems 123 (2001) 271–290.
[40] Z. Quin, J. Lawry, Decision tree learning with fuzzy labels, Information Sciences 172 (2005) 91–129.
[41] J.R. Quinlan, Induction of decision trees, Machine Learning 1 (1986) 81–106.
[42] J.R. Quinlan, Decision trees as probabilistic classifiers, in: Proceedings of the 4th International Workshop on Machine Learning, 1987, pp. 31–37.
[43] J.R. Quinlan, C4.5: Programs for Machine Learning, Springer, 1993.
[44] C.E. Shannon, The mathematical theory of communication, The Bell system Technical Journal 27 (3) (1948) 379–423.
[45] P. Smets, R. Kennes, The transferable belief model, Artificial Intelligence 66 (1994) 191–234.
[46] S. Trabelsi, Z. Elouedi, K. Mellouli, Pruning belief decision tree methods in averaging and conjunctive approaches, International Journal of Approximate Reasoning (2007), doi:10.1016/j.ijar.2007.02.004.
[47] P. Vannoorenbergue, T. Denoeux, Handling uncertain labels in multiclass problems using belief decision trees, in: Proceedings of the Ninth International Conference on Information processing and Management of Uncertainty in Knowledge-Based systems (IPMU'02), vol. III, Annecy, France, July 2002, pp. 1919–1926.
[48] P. Walley, Statistical Reasoning with Imprecise Probabilities, Chapman and Hall, New York, 1991.
[49] S.M. Weiss, C.A. Kulikovski, Computer Systems that Learn, Morgan Kaufman, San Mateo, CA, 1991.
[50] I.H. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, second ed., Morgan Kaufman publishers, 2005.
[51] R.R. Yager, On the specificity of a possibility distribution, Fuzzy Sets and Systems 50 (1992) 279–292.
[52] Y. Yuan, M.J. Shaw, Induction of fuzzy decision trees, Fuzzy Sets and Systems 69 (1995) 125–139.
[53] L.A. Zadeh, Fuzzy sets as a basis for a theory of possibility, Fuzzy Sets and Systems 1 (1978) 3–28.
[54] M. Zemankova, A. Kandel, Implementing imprecision in information systems, Information Sciences 37 (1–3) (1985) 07–141.