entropy

MDPI

Article

Belief Entropy Tree and Random Forest: Learning from Data with Continuous Attributes and Evidential Labels

Kangkai Gao [1], Yong Wang [1,*] and Liyao Ma [2]

[1] Department of Automation, University of Science and Technology of China, Hefei 230027, China; gkk2010@mail.ustc.edu.cn
[2] School of Electrical Engineering, University of Jinan, Jinan 250022, China; cse_maly@ujn.edu.cn
* Correspondence: yongwang@ustc.edu.cn; Tel.: +86-0551-6360-1506

Abstract: As well-known machine learning methods, decision trees are widely applied in classification and recognition areas. In this paper, with the uncertainty of labels handled by belief functions, a new decision tree method based on belief entropy is proposed and then extended to random forest. With the Gaussian mixture model, this tree method is able to deal with continuous attribute values directly, without pretreatment of discretization. Specifically, the tree method adopts belief entropy, a kind of uncertainty measurement based on the basic belief assignment, as a new attribute selection tool. To improve the classification performance, we constructed a random forest based on the basic trees and discuss different prediction combination strategies. Some numerical experiments on UCI machine learning data set were conducted, which indicate the good classification accuracy of the proposed method in different situations, especially on data with huge uncertainty.

Keywords: decision trees; uncertain data; belief entropy; belief function; random forest; evidential likelihood

Citation: Gao, K.; Wang, Y.; Ma, L. Belief Entropy Tree and Random Forest: Learning from Data with Continuous Attributes and Evidential Labels. *Entropy* **2022**, *24*, 605. https://doi.org/10.3390/e24050605

Academic Editors: Andrea Prati, Luis Javier García Villalba and Vincent A. Cicirello

Received: 26 March 2022
Accepted: 23 April 2022
Published: 26 April 2022

1. Introduction

Decision trees have been widely used for their good learning capabilities and ease of understanding. In some real world issues, instances may be ill-known for some factors such as randomness, data incompleteness and even expert's indefinite subjective opinions; however, traditional decision trees can only handle certain samples with precise data. The incompletely observed instances are usually ignored or replaced by a precise one, despite the fact that they may contain useful information [1], which may cause a loss of accuracy.

There have been many attempts to build trees from incomplete data in the past several decades. The probability trees [2,3] were suggested based on probability theory, which is usually intuitively the first tool to modeling uncertainty in practice; however, it has been proven that probability cannot always be adequate for representing data uncertainty [4,5] (often termed epistemic uncertainty). To overcome this drawback, various approaches have been proposed, including: fuzzy decision trees [6,7], the possibilistic decision trees [8] and the uncertain decision trees [9,10]. Besides the aforementioned methods, a more general framework, called the belief function theory [11,12] (also evidential theory or Dempster–Shafer theory), has been proven to have the ability to model all kinds of knowledge. The process of embedding belief functions within decision tree techniques has already been extensively investigated [13–25] in recent years. Particularly, among these methods, several trees [17–19] estimate parameters by maximizing evidential likelihood function using the E^2M algorithm [26,27], which is also the basis of part of the trees to be proposed in this paper.

However, the existing methods on incomplete data do not take continuous attributes into full consideration. These proposals deal with uncertain data modeled by the belief function and build trees by extending the traditional decision tree method. The imitation and transformation decides to use existing methods to handle continuous attribute values

by discretization, which brings about an issue of losing the detail of the training data. For example, the information gain ratio, the attribute selecting measurement in C4.5, was transformed to adapt the evidential labels of the training set in the Belief C4.5 trees [19], in which the continuous-valued attribute is divided into four intervals of equal width before learning. This issue leads to the purpose of this paper: to learn from uncertain data with continuous attribute values without pretreatment.

To realize this purpose, we firstly, for each attribute, fit the training data to a Gaussian mixture model (GMM), which consists of normal distribution models one-by-one corresponding to class labels, by adopting the E^2M algorithm. This step, which significantly differs from other decision trees, confirms the ability to deal with ill-known labels and original attribute values (either discrete or continuous). On the basis of these GMM models, we generate the basic belief assignment (BBA) and calculate belief entropy [28]. The attribute with minimal average entropy, which distinguishes classes from each others most, will be selected as the splitting attribute. The following decision tree induction steps are designed accordingly and logically. To our knowledge, this paper is the first to introduce GMM models and belief entropy to decision trees with evidential data.

Another part of our proposal is adopting the ensemble method for our belief entropy trees. Inspired by the idea of building bagging trees based on random sampling [29], we further choose a more efficient and popular technique—random forest [30]. Under the belief function framework, the basic trees will output either precise or mass (modeled by BBA) label predictions, while traditional random forest can only combine precise labels. Thus, a new method to summarize the basic tree predictions is proposed to combine mass labels directly, instead of voting on precise labels. This combined mass keeps the uncertain information of data as much as possible, which helps to generate a more reasonable prediction. The new combination method is discussed and compared to the traditional majority voting method later.

We note that we have proposed our early work in a shorter conference paper [31]. Compared with our initial conference paper, we have fixed the attribute selection and splitting strategy of a single tree and introduced ensemble learning to our tree method in this paper.

Section 2 recalls some basic knowledge about decision trees, belief function theory, the E^2M algorithm and belief entropy. Section 3 details the induction procedure of belief entropy methods and proposes three different instance prediction techniques. In Section 4, we introduce how to expend the single belief entropy tree to random forests and discuss the different predicting combination strategies. In Section 5, we detail experiments that were carried out on some classical UCI machine learning data sets to compare the classification accuracies of proposed trees and random forests. Finally, conclusions are summarized in Section 6.

2. Settings and Basic Definitions

The purpose of a classification method is to build a model that maps an attribute vector $X = (x^1, \ldots, x^D) \in A^1 \times A^2 \times \cdots \times A^D$, which contains D attributes, to an output class $y \in \mathcal{C} = \{C_1, \ldots, C_K\}$ taking its value among K classes. Each attribute discretely has finite values or continuously takes value within an interval. The learning of classification is based on a complete training set of precise data which contains N instances, denoted as

$$T = \begin{pmatrix} X_1, y_1 \\ \vdots \\ X_N, y_N \end{pmatrix} = \begin{pmatrix} x_1^1, \ldots, x_1^D, y_1 \\ \vdots \\ x_N^1, \ldots, x_N^D, y_N \end{pmatrix}.$$

However, the imperfect knowledge about the inputs (feature vector) and the outputs (classification labels) exists widely in practical applications. Traditionally and regularly, the imperfect knowledge is modeled by probability theory, which is considered to be questionable in a variety of scenarios. Hence, we model uncertainty by belief function

in this paper. Typically, we consider that attribute values are precise and can be either continuous or discrete, while only the output labels are uncertain.

2.1. Decision Trees

Decision trees [32] are regarded as one of the most effective and efficient machine learning methods and widely adopted for solving classification and regression problems in practice. The success, to a great extent, relays on the easily understandable structure, for both humans and computers. Generally, a decision tree is induced top-down from a training set T, which recursively repeats the steps below:

- Select an attribute, through a designed selection method, to generate a partition of a training set;
- Split the current training set to several subsets and put them into child node;
- Generate a leaf node and determine the prediction label for a child node when a stop criterion is satisfied.

Differing in the attribute selection methods, several decision tree algorithms have been proposed, such as ID3 [32], C4.5 [33] and CART [34]. Among these trees, the ID3 and C4.5 choose entropy as an information measure to compute and evaluate the quality of a node split by a given attribute.

The core of ID3 is information gain. Given training data T and an attribute A with K_A modalities, the information gain will be:

$$Gain(T, A) = Info(T) - Info_A(T) \tag{1}$$

where

$$Info(T) = -\sum_{i=1}^{K} \theta_i \log_2(\theta_i) \tag{2}$$

and

$$Info_A(T) = -\sum_{i=1}^{K_A} \frac{|T_i|}{|T|} Info(T_i) \tag{3}$$

where θ_i is the proportion of instances in T that are of class C_i, $|T|$ and $|T_i|$ are the cardinalities of the instance sets belonging to a parent node and to the child node i.

The limitation of information gain is that attributes with largest values will be most promoted [33], which leads to the *GainRatio* in the C4.5 algorithm. It is given as:

$$GainRatio(T, A) = \frac{Gain(T, A)}{SplitInfo(T, A)} \tag{4}$$

where

$$SplitInfo(T, A) = -\sum_{i=1}^{K_A} \frac{|T_i|}{|T|} \log_2 \frac{|T_i|}{|T|}. \tag{5}$$

The attribute with the largest gain ratio will be selected for splitting.

We can easily find the the Equation (2) is actually the Shannon Entropy. Yet in this paper, concerning the feature of evidential data described by the framework of belief function, the attribute selection method is newly designed based on belief entropy [28] instead of Shannon entropy.

2.2. Random Forest

To improve the classification accuracy and generalization ability of machine learning, the ensemble model method is introduced to the learning procedure. One important branch of ensemble method is called *bagging*, which concurrently builds multiple basic models learning from different training sets, which are generated from original data by bootstrap sampling. On the basis of bagging decision trees, random forest (RF) [30] not only chooses the training instance randomly but also introduces randomness into attributes selection. To

be specific, traditional decision trees select the best splitting attribute among all D attributes; random forest generates a random attribute subset then chooses the best one within this subset to split the tree node. The size D' of this subset is adjustable and generally set as $D' = log_2 D$.

A detailed description of the mathematical formulation of RF model is found in [30]. The RF model consists of a union of multiple basic trees, where each tree learns from bootstrap samples and selects attribute from a small subset of all attributes. There some advantages of RF: (a) better prediction performance, (b) resistance to overfitting, (c) low correlation of individual trees, (d) low bias and low variance and (e) small computational overhead.

Some existing works have explored the ensemble method on belief decision trees, such as bagging [29]. In this paper, we apply the random forest technique to the proposed belief entropy trees and discuss the different prediction determining strategies.

2.3. Belief Function Theory

Let the finite set Ω denote the frame of discernment containing k possible exclusive values that a variable can take. When considering the output y, the imperfect knowledge about value of y can be modeled by mass function $m_y : 2^\Omega \to [0,1]$, such that $m_y(\varnothing) = 0$, and

$$\sum_{A \subseteq \Omega} m_y(A) = 1, \tag{6}$$

which is also called a basic belief assignment (BBA). The subset A is called a focal set where $m_y(A) > 0$, and the $m_y(A)$ can be interpreted as the support degree of the evidence towards the case that true value is in set A.

There are some typical mass functions need to be attended:

- *Vacuous* mass: mass function such that $m_y(\Omega) = 1$, which means total ignorance;
- *Bayesian* mass: for all focal set A, the cardinality $|A| = 1$. In this case, the mass degenerates to a probability distribution;
- *Logical*$(categorical)$ mass: $m_y(A) = 1$ for some A. In this case, the mass is equivalent to the set A.

One-to-one related to the mass function m_y, the $belief\ function$ and $plausibility\ function$ are defined as:

$$Bel_y(B) = \sum_{A \subseteq B} m_y(A), \tag{7}$$

$$Pl_y(B) = \sum_{A \cap B \neq \varnothing} m_y(A), \tag{8}$$

which, respectively, indicate the minimum and maximum belief degree of evidence towards set B. Typically, the function $pl : \Omega \to [0,1]$ such that $pl_y(\omega) = Pl_y(\{\omega\})$ for all $\omega \in \Omega$ is called $contour\ function$ associated to m_y.

For two mass function m_1 and m_2 induced by evidences independently, they can be combined by the *Dempster's rule* [12] $\oplus$ defined as:

$$(m_1 \oplus m_2)(A) = \frac{1}{1-\kappa} \sum_{B \cap C = A} m_1(B) m_2(C), \tag{9}$$

for all $A \subseteq \Omega$, $A \neq \varnothing$, and $(m_1 \oplus m_2)(\varnothing) = 0$, where

$$\kappa = \sum_{B \cap C = \varnothing} m_1(B) m_2(C), \tag{10}$$

is called the $degree\ of\ conflict$ between m_1 and m_2. Obviously, Dempster's rule is commutative and associative according to the definition.

In the decision making situation, we need to determine the most reasonable hypothesis from a mass. Different decision-making strategies with belief functions [35,36]

have been researched. Among these methods, in the transferable belief model (TBM), *pignistic probability* [37] was proposed to make decision from a BBA:

$$BetP(\omega) = \sum_{A \subseteq \Omega, \omega \in A} \frac{m(A)}{|A|}, \tag{11}$$

where $|A|$ is the cardinality of A.

When we model uncertain labels of evidential data with mass functions, the training set becomes

$$T = \begin{pmatrix} X_1, m_1 \\ \vdots \\ X_N, m_N \end{pmatrix} = \begin{pmatrix} x_1^1, \ldots, x_1^D, m_1 \\ \vdots \\ x_N^1, \ldots, x_N^D, m_N \end{pmatrix}.$$

2.4. Evidential Likelihood

Consider a discrete random vector Y taking values in Ω with a probability mass function $p_Y(y; \boldsymbol{\theta})$ assumed to be associated with a parameter $\boldsymbol{\theta} \in \Theta$. After a realization y of Y has been perfectly observed, the likelihood function of *complete data* is defined as $L : \Theta \to [0, 1]$ such that

$$L(\boldsymbol{\theta}; y) = p_Y(y; \boldsymbol{\theta}), \forall \boldsymbol{\theta} \in \Theta. \tag{12}$$

When the observations are uncertain, it is impossible to evaluate parameter $\boldsymbol{\theta}$ from a likelihood function. In this situation, a new statistical tool [38] called evidential likelihood was proposed to perform parameter estimation. Assume that y is not precisely observed, but is known surely that $y \in A$ for some $A \in \Omega$. Given such *imprecise data*, the likelihood function will be extended to

$$L(\boldsymbol{\theta}; A) = p_Y(A; \boldsymbol{\theta}) = \sum_{y \in A} p_Y(y; \boldsymbol{\theta}), \forall \boldsymbol{\theta} \in \Theta. \tag{13}$$

Furthermore, the observation of instance y could be not only *imprecise*, but also *uncertain*, which is modeled by mass function m_y. Thus the evidential likelihood function [27] can be defined as

$$\begin{aligned} L(\boldsymbol{\theta}; m_y) &= \sum_{A \subseteq \Omega} L(\boldsymbol{\theta}; A) m_y(A) \\ &= \sum_{y \in \Omega} p_Y(y; \boldsymbol{\theta}) \sum_{A \ni x} m_y(A) \\ &= \sum_{y \in \Omega} p_Y(y; \boldsymbol{\theta}) pl(y), \forall \boldsymbol{\theta} \in \Theta \end{aligned}, \tag{14}$$

where the pl is the contour function related to m_y and the $L(\boldsymbol{\theta}; m_y)$ can be remarked as $L(\boldsymbol{\theta}; pl)$. According to the statement of Denoeux [27], the value $1 - L(\boldsymbol{\theta}; pl)$ equals to the conflict between parametric model $p_Y(y; \boldsymbol{\theta})$ and the uncertain observation pl, which means minimizing $L(\boldsymbol{\theta}; pl)$ is actually a procedure of estimating the best parameter $\boldsymbol{\theta}$ to fit the parametric model to observation as closely as much.

Equation (14) also indicates that $L(\boldsymbol{\theta}; pl)$ can be remarked as the expectation of pl such that

$$L(\boldsymbol{\theta}; pl) = \mathbb{E}_{\boldsymbol{\theta}}[pl(Y)]. \tag{15}$$

Assume that $Y = (y_1, \ldots, y_N)$ is a sample set containing n cognitively independent [12] and i.i.d. uncertain observations, in which the y_i is model by m_{y_i}. In the situation the Equation (15) is written as a product of n terms:

$$L(\boldsymbol{\theta}; pl) = \prod_{n=1}^{N} \mathbb{E}_{\boldsymbol{\theta}}[pl_n(y_n)]. \tag{16}$$

2.5. E^2M Algorithm

Though an extension of likelihood function, the maximum likelihood estimation of evidential likelihood can not directly be computed by the broadly applied EM algorithm [39]. The E^2M algorithms [27] introduced by Denoeux allow us to maximize the evidential likelihood iteratively, which is composed of two steps (similar to EM algorithm):

1. The **E-step** require firstly a probability mass function $p_Y\left(\cdot \mid pl; \boldsymbol{\theta}^{(q)}\right) = p_Y\left(\cdot; \boldsymbol{\theta}^{(q)}\right) \oplus pl$, in which the former part means the probability mass function of Y under the parameter $\boldsymbol{\theta}^{(q)}$ estimated from last iteration and the latter part indicates contour function pl. The expression is:

$$p_Y\left(y \mid pl; \boldsymbol{\theta}^{(q)}\right) = \frac{p_Y\left(y; \boldsymbol{\theta}^{(q)}\right) pl(y)}{L\left(\boldsymbol{\theta}^{(q)}; pl\right)}. \tag{17}$$

   Then calculate the expectation of log likelihood $\log L_c(\boldsymbol{\theta}; y) = \log p_Y(y; \boldsymbol{\theta})$ of complete data with respect to $p_Y\left(\cdot \mid pl; \boldsymbol{\theta}^{(q)}\right)$,

$$Q\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{(q)}\right) = \frac{\sum_{y \in \Omega} \log(L_c(\boldsymbol{\theta}; y)) p_Y\left(y; \boldsymbol{\theta}^{(q)}\right) pl(y)}{L\left(\boldsymbol{\theta}^{(q)}; pl\right)}. \tag{18}$$

2. The **M-step** is to maximize $Q\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{(q)}\right)$ with respect to $\boldsymbol{\theta}$, obtaining a new estimation that ensures $Q\left(\boldsymbol{\theta}^{(q+1)}, \boldsymbol{\theta}^{(q)}\right) \geqslant Q\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{(q)}\right)$.

   The two steps repeat until $L\left(\boldsymbol{\theta}^{(q+1)}\right) - L\left(\boldsymbol{\theta}^{(q)}\right) \leqslant \epsilon$, where ϵ is a set threshold.

2.6. Belief Entropy

Inspired by Shannon entropy [40], which can measure uncertainty contained by a probability distribution, a type of belief entropy called Deng entropy is proposed by Deng [28] to handle situation where the traditional probability theory is limited. When the uncertain information is described by the basic belief assignment instead of the probability distribution, Shannon entropy cannot work. Deng entropy is defined on the belief function frame, which makes it able to measure uncertain information described by the BBA efficiently.

Let A be the focal set of belief function, and $|A|$ be the cardinality of A. Deng entropy E is defined as:

$$E(m) = -\sum_{A \subseteq \Omega} m(A) \log \frac{m(A)}{2^{|A|} - 1}. \tag{19}$$

We can easily learn from the definition that if the mass function is *Bayesian*, which means $|A| = 1$ for all A, Deng entropy degenerates to Shannon entropy such that

$$E(m) = -\sum_{A \subseteq \Omega} m(A) \log m(A). \tag{20}$$

The greater the cardinality of the focal set is, the bigger the corresponding Deng entropy is, so that the evidence imprecisely refers to more single elements. Thus, significant Deng entropy indicates huge uncertainty. Powered by this feature, we calculate the average Deng entropy of BBAs to select the best attribute leading to the least uncertainty. The details are shown in the next section.

3. Design of Belief Entropy Trees

Up to now, various decision tree methods have been proposed to deal with evidential data, but many of them consider categorical attributes and transform the continuous attribute values into discrete categories. Some recent works fit the continuous attributes

with same class labels into normal distributions [41] and generate BBA from the normal distributions to select the best splitting attribute by calculating belief entropy [42]; however, this method divides samples into each set of certain classes, which can only handle the precise class labels. Our goal is to develop a belief decision tree method learns from data set with continuous and precise attribute values but incomplete class labels directly and efficiently.

This section explains our method in detail, specifically focusing on the procedure of attribute selection. Corresponding splitting strategy, stopping criterion and the leaf structure are also well-designed to accomplish the whole belief entropy decision tree.

3.1. The Novel Method to Select Attribute

The learning procedure of decision trees is generally to decide the split attribute and to decide how to split on this attribute on each node; our method also proceeds in this manner. As a novel decision tree, the most characteristic and core part of our method is the attribution selection, which includes three steps: firstly, for each attribute, fit the values to normal distributions corresponding to each class label, in another words, fit attribute values into K×D normal distribution models, where K is the class number and D is the attributes number of instances; secondly, for every instance, generate D BBAs from each attribute according to the normal distribution-based models; finally, calculate belief entropy from BBAs for each attribute. The attribute with minimum belief entropy will be selected to split.

3.1.1. Parameter Estimation on Data with Continuous Attributes

Powered by the idea of extracting BBAs from normal distribution-modeled attribute values [41], we try to operate similarly on data with ill-known class labels. In the situation that each instance exactly belongs to one class, the d-th attribute values set $\left\{x_1^d, \ldots, x_N^d\right\}$ is divided into K subsets $\left\{x_n^d \mid y_n = C_k\right\}, k = 1, \ldots, K$ corresponding to each class. It is easy to fit each subset to the normal distribution by calculating means and standard deviations.

Example 1. *Consider the Iris data set [43], a classical machine learning data set, which contains 150 training instances of three classes: 'Setosa', 'Versicolor', 'Virginica', with four attributes: sepal length(SL), sepal width(SW), petal length(PL) and petal width(PW). For the values of attribute SL in the class of Setosa, we can directly calculate the mean value as* $\mu = 5.0133$ *and standard deviation as* $\sigma = 0.3267$*. Similarly, we can obtain normal distribution parameters of class of Versicolor and Virginica. Figure 1 shows the normal distribution model of Iris data set for the SL attribute in three classes.*

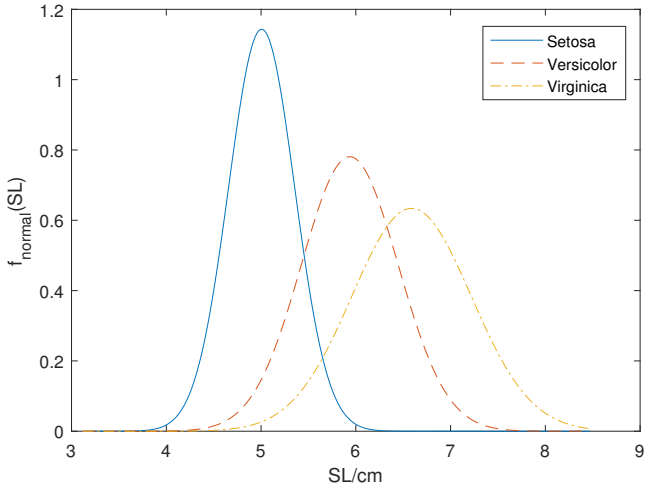


Figure 1. The normal distribution df three classes for the SL attribute of Iris data set.

However, when the labels of training set are ill-known, some instances can not be allocated to a certain class assertively. The evidential likelihood and E^2M algorithm introduced in Section 2 make it possible to generate an estimation of model parameters. Because the E^2M algorithm uses only contour functions, the label of n-th instance will be represented by plausibility $pl_n = \{pl_{nk}\}, k = 1, \ldots, K$ instead of mass function m_n.

For the purpose of comparing attributes, we split the whole training data into D attribute–label pairs and handle D parameter estimation problems. Consider the d-th attribute value vector $X^d = \left(x_1^d, \cdots, x_N^d\right)^T, d \in \{1, \ldots, D\}$, we assume the conditional distribution of X^d when given $y = C_k$ is normal with mean μ_k and standard deviation σ_k:

$$X^d | (y = C_k) \sim \mathcal{N}\left(\mu_k, \sigma_k^2\right), k = 1, \ldots, K.$$

Actually the assumption is to build a one-dimensional *Gaussian mixture model*(GMM) [44]. Similar to the application of E^2M algorithm in *linear discriminant analysis* [45], the following discuss is practically to adopt E^2M algorithm to estimate parameters in GMM.

Let π_k be the marginal probability when $y = C_k$, and $\theta = (\mu_1, \ldots, \mu_K, \sigma_1, \ldots, \sigma_K, \pi_1, \ldots, \pi_K)$ the parameter vector to be estimated. The complete-data likelihood is

$$\begin{aligned} L_c(\theta) &= \prod_{n=1}^{N} p\left(x_n^d | Y_n = y_n\right) p(y_n) \\ &= \prod_{n=1}^{N} \prod_{k=1}^{K} \phi\left(x_n^d; \mu_k, \sigma_k\right)^{y_{nk}} \pi_k^{y_{nk}}, \end{aligned} \tag{21}$$

where the ϕ is normal distribution probability density,

$$\phi(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \tag{22}$$

and y_{nk} is a binary indicator variable, such that $y_{nk} = 1$ if $y_n = C_k$ and $y_{ik} = 0$ if $y_N \neq C_k$.

when expended to evidential data, where we use contour function to describe the labels, the evidential likelihood is drew from Equation (16) that,

$$\begin{aligned} L(\theta) &= \prod_{n=1}^{N} \mathbb{E}_{\boldsymbol{\theta}}[pl_n(y_n)] \\ &= \prod_{n=1}^{N} \sum_{k=1}^{K} pl_{nk} \phi\left(x_n^d; \mu_k, \sigma_k\right) \pi_k. \end{aligned} \tag{23}$$

According to the E^2M algorithm, we compute the expectation of complete-data log likelihood

$$\ell_c(\theta) = \log L_c(\theta) = \sum_{n=1}^{N} \sum_{k=1}^{K} y_{nk} \left[\log \phi\left(x_i^d; \mu_k, \sigma_k\right) + \log \pi_k\right] \tag{24}$$

with respect to the combined mass probability function

$$p\left(x^d \middle| pl; \theta^{(q)}\right) = \prod_{n=1}^{N} p\left(x_n^d \middle| pl_n; \theta^{(q)}\right). \tag{25}$$

To simplify the equation, we denote

$$\zeta_{nk}^{(q)} = p\left(x_n^d \middle| pl_n; \theta^{(q)}\right) = \frac{pl_{nk} \pi_k^{(q)} \phi\left(x_n^d; \mu_k^{(q)}, \sigma_k^{(q)}\right)}{\sum_{k=1}^{K} pl_{nk} \pi_k^{(q)} \phi\left(x_n^d; \mu_k^{(q)}, \sigma_k^{(q)}\right)}. \tag{26}$$

Finally, we obtain the to-be-maximized function

$$Q\left(\theta,\theta^{(q)}\right)=\sum_{n=1}^{N}\sum_{k=1}^{K}\zeta_{nk}^{(q)}\left[\log\phi\left(x_n^d;\mu_k,\sigma_k\right)+\log\pi_k\right] \tag{27}$$

in the E-step.

The formal of $Q\left(\theta,\theta^{(q)}\right)$ is similar to the function computed in the EM algorithm on the GMM [44]. Because of the similarity, we imitate it and learn that the optimal parameter maximizing $Q\left(\theta,\theta^{(q)}\right)$ can be iteratively computed by

$$\pi_k^{(q+1)}=\frac{1}{n}\sum_{n=1}^{N}\zeta_{nk}^{(q)},\ \mu_k^{(q+1)}=\frac{\sum_{n=1}^{N}\zeta_{nk}^{(q)}x_i^d}{\sum_{n=1}^{N}\zeta_{nk}^{(q)}}, \tag{28}$$

$$\sigma_k^{(q+1)}=\sqrt{\frac{\sum_{n=1}^{N}\zeta_{nk}^{(q)}\left(x_n^d-\mu_k^{(q+1)}\right)^2}{\sum_{n=1}^{N}\zeta_{nk}^{(q)}}} \tag{29}$$

Finally when $L\left(\boldsymbol{\theta}^{(q+1)}\right)-L\left(\boldsymbol{\theta}^{(q)}\right)\leqslant\epsilon$ is satisfied for some ϵ, stop the iteration and remark $\theta^{(q+1)}$ as $\theta^d=\left(\mu_1^d,\ldots,\mu_K^d,\sigma_1^d,\ldots,\sigma_K^d,\pi_1^d,\ldots,\pi_K^d\right)$, which is the estimation of parameters in the GMM extracted from d-th attribute. Repeat this procedure for every attributes of the training set, $D\times K$ normal distribution

$$\mathcal{N}_k^d\left(\mu_k^d,\sigma_k^{d2}\right),d=1,\ldots,D,k=1,\ldots,K$$

will be generated.

The Algorithm 1 shows the procedure of parameter estimation and there is Example 2 to help understand it.

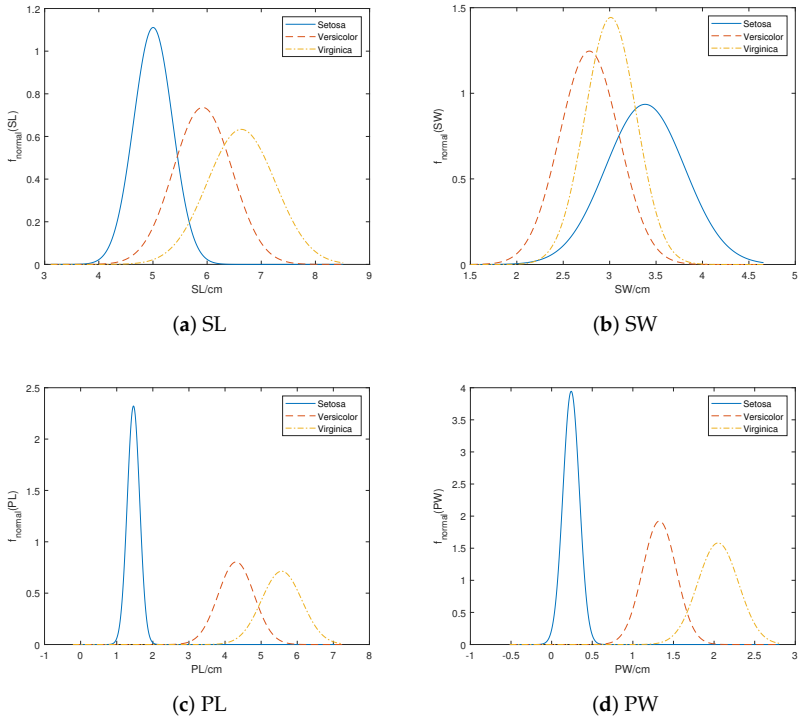
Algorithm 1 Parameter estimation of GMMs.

Input: evidential training set $T_{pl}=(\boldsymbol{x},pl_y)$, iteration stop threshold ϵ

Output: estimated normal distribution parameter matrix $\left(\mu_k^d,\sigma_k^d\right),d=1,\ldots,D,k=1,\ldots,K$

for each attribute A^d **do**
  initialize parameters as $\theta^{d(0)}=\left(\mu_1^{d(0)},\ldots,\mu_K^{d(0)},\sigma_1^{d(0)},\ldots,\sigma_K^{d(0)},\pi_1^{d(0)},\ldots,\pi_K^{d(0)}\right)$;
  $q=0$; {Initialize loop variable.}
  for q **do**
    update the estimation of parameters $\theta^{d(q+1)}=\left(\mu_1^{d(q+1)},\ldots,\mu_K^{d(q+1)},\sigma_1^{d(q+1)},\ldots,\sigma_K^{d(q+1)},\pi_1^{d(q+1)},\ldots,\pi_K^{d(q+1)}\right)$ by Equations (28) and (29).
    if $L\left(\theta^{d(q+1)}\right)-L\left(\theta^{d(q)}\right)\leqslant\epsilon$ **then**
      break; {End the loop if evidential likelihood increment is less than threshold.}
    end if
    $q=q+1$;
  end for
  adopt $\left(\mu_k^{d(q+1)},\sigma_k^{d(q+1)}\right),k=1,\ldots,K$ as estimated normal distribution parameters under attribute A^d;
end for

Example 2. *Consider the Iris data set mentioned in Example 1. To simulate the situation that labels of training set are not completely observed, we manually introduce uncertainty to the Iris data. In this example, we set that each instance has an equivalent chance (25%) to be vacuous, imprecise, uncertain or completely observed (the detail of transformation is discussed in Section 5). Table 1 shows the attribute values and labels described by plausibility pl of some instances in evidential Iris data. Table 2 shows the mean and standard deviation pairs* (μ, σ) *calculated by* E^2M *algorithm. Figure 2 shows curves of these models.*



(**a**) SL (**b**) SW

(**c**) PL (**d**) PW

Figure 2. The normal distribution models for each attribute in evidential Iris data set.

Table 1. Uncertain instances in evidential Iris data set.

Number	Attributes				Contour Functions			True Label
	SL	SW	PL	PW	pl (Setosa)	pl (Versicolor)	pl (Virginica)	
1	5.1	3.5	1.4	0.2	1	0	1	Setosa
2	4.9	3.0	1.4	0.2	1	0	1	Setosa
3	4.7	3.2	1.3	0.2	1	0	0	Setosa
4	4.6	3.1	1.5	0.2	1	0.7498	0.4073	Setosa
...								
51	7.0	3.2	4.7	1.4	0	1	0	Versicolor
52	6.4	3.2	4.5	1.5	0.9519	1	0.7087	Versicolor
53	6.9	3.1	4.9	1.5	1	1	1	Versicolor
54	5.5	2.3	4.0	1.3	1	1	1	Versicolor
...								
101	6.3	3.3	6.0	2.5	0	0	1	Virginica
102	5.8	2.7	5.1	1.9	0.4458	0.5088	1	Virginica
103	7.1	3.0	5.9	2.1	0	0	1	Virginica
104	6.3	2.9	5.6	1.8	1	1	1	Virginica
...								
150	5.9	3.0	5.1	1.8	0	0	1	Virginica

Table 2. Estimated normal distribution parameters for evidential Iris data set.

Attributes	Setosa		Versicolor		Virginica	
	μ	σ	μ	σ	μ	σ
SL	5.0031	0.3591	5.9183	0.5419	6.6307	0.6300
SW	3.3822	0.4264	2.7812	0.3201	3.0119	0.2765
PL	1.4640	0.1718	4.3066	0.4961	5.5684	0.5591
PW	0.2408	0.1011	1.3273	0.2074	2.0495	0.2522

3.1.2. BBA Determination

This step is to generate D BBAs corresponding to each attribute for every instance in the training set.

Choose an instance I_n with attribute vector $x_n = \left(x_n^1, \ldots, x_n^D\right)$ from the data set, calculate the intersection of $x_n^d (d = 1, \ldots, D)$ and the K normal distribution functions $\phi_k^d = \phi\left(x^d; \mu_k, \sigma_k\right), k = 1, \ldots, K$, i.e., we obtain K normally distributed probability density function (PDF) values for the attribute A^d and instance I_n, denoted as $\phi_{nk}^d, k = 1, \ldots, K$.

Due to the property that the probability of a value x sampling from a normal distribution is proportional to the PDF $\phi(x)$, we can infer, for the attribute d, the probability that instance x_n belongs to each class is proportional to $\phi_{nk}^d = \phi\left(x_n^d; \mu_k, \sigma_k\right), k = 1, \ldots, K$. From this opinion of statistical analysis, the rule to assign normal PDFs to some sets was proposed to build BBAs.

Firstly, normalize the ϕ_{nk}^d with different class k such that

$$f_k = \phi_{nk}^d / \sum_{k=1}^{K} \phi_{nk}^d. \tag{30}$$

Then rank f_k in decreasing order $f_r' (r = 1, \ldots, K)$, whose corresponding class is denoted as $C_r' (r = 1, \ldots, K)$. Assign f_r' to the class set by the following rule:

$$\begin{aligned} &m\left(\left\{C_1'\right\}\right) = f_1' \\ &m\left(\left\{C_1', C_2'\right\}\right) = f_2' \\ &\ldots \\ &m\left(\left\{C_1', \ldots, C_K'\right\}\right) = m(\theta) = f_K'. \end{aligned} \tag{31}$$

If $f_i' = f_{i+1}' = \ldots = f_j'$, then $m\left(\left\{C_1', \ldots, C_j'\right\}\right) = \sum_{p=i}^{j} f_p'$. By this rule, we obtain a nested BBA of x_n under the select attribute A^d, which we denote as m_n^d.

Example 3. *Consider the first instance of the evidential Iris data set showed in Table 1, whose attributes are:*

$$x^{SL} = 5.1,\ x^{SW} = 3.5,\ x^{PL} = 1.4,\ X^{PW} = 0.2.$$

For attribute SL, the intersections of $x^{SL} = 5.1$ *and three normal distributions are shown in Figure 3 such that*

$$\begin{aligned} &\phi_{Setosa}^{SL}(x^{SL}) = 1.0712, \\ &\phi_{Versicolor}^{SL}(x^{SL}) = 0.2354, \\ &\phi_{Virginica}^{SL}(x^{SL}) = 0.0331. \end{aligned}$$

The reader can see in the figure that this instance is closest to class 'Setosa', then to the 'Versicolor' and 'Virginica'. Thus, we generate BBA from intersection values, which is intuitive. The BBA is assigned as:

$$m(\{Setosa\}) = \frac{1.0712}{1.0712 + 0.2354 + 0.0331} = 0.7996$$
$$m(\{Setosa, Versicolor\}) = \frac{0.2354}{1.0712 + 0.2354 + 0.0331} = 0.1757$$
$$m(\{Setosa, Versicolor, Virginica\}) = \frac{0.0331}{1.0712 + 0.2354 + 0.0331} = 0.0247$$

Similarly, we build BBAs for the rest of the attributes—shown in Table 3.

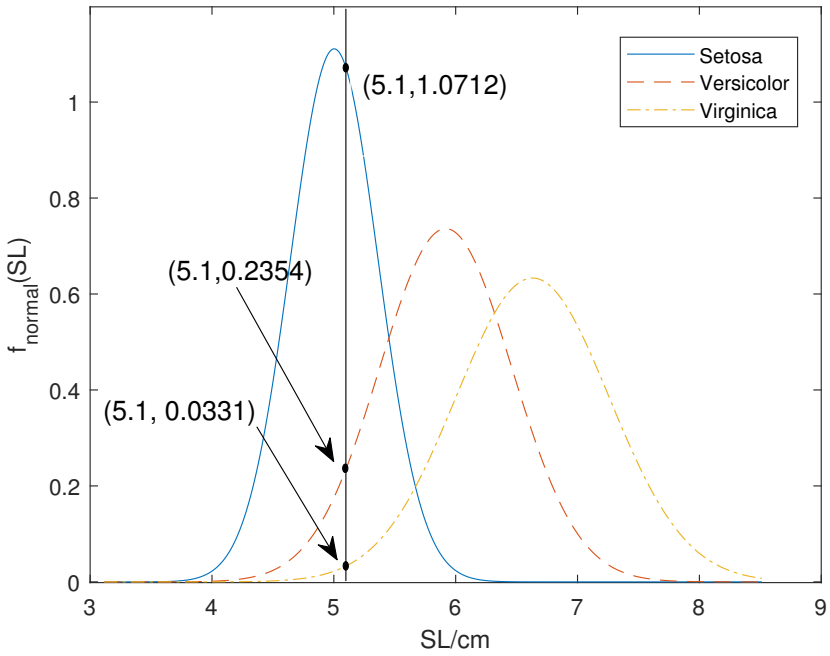


Figure 3. The normal distribution for the SL attribute in three classes.

Table 3. Generated BBAs of selected instance.

Attributes	BBAs
SL	$m(\{Setosa\}) = 0.7996$ $m(\{Setosa, Versicolor\}) = 0.1757$ $m(\{Setosa, Versicolor, Virginica\}) = 0.0247$
SW	$m(\{Setosa\}) = 0.6904$ $m(\{Setosa, Virginica\}) = 0.2329$ $m(\{Setosa, Versicolor, Virginica\}) = 0.0767$
PL	$m(\{Setosa\}) = 1$ $m(\{Setosa, Versicolor\}) = 0$ $m(\{Setosa, Versicolor, Virginica\}) = 0$
PW	$m(\{Setosa\}) = 1$ $m(\{Setosa, Versicolor\}) = 0$ $m(\{Setosa, Versicolor, Virginica\}) = 0$

3.1.3. Calculation of Belief Entropy

The last step to determine splitting attribute is to calculate the average Deng entropy

$$E^d = E\left(A^d\right) = \frac{1}{N}\sum_{n=1}^{N} E\left(m_n^d\right), d = 1, \ldots D \tag{32}$$

of all instances for each attribute. As mentioned in Section 2.6, Deng entropy measures the uncertain degree contained by BBA, which means the less E^d, the more certainty the BBAs contain, and the more separate the division of classes is. Consequently, we choose the attribute A^* that minimizes the average Deng entropy such that

$$A^* = \arg\min_{A^d}\left(E\left(A^d\right)\right), d = 1, \ldots, D \tag{33}$$

to be the best splitting attribute to proceed the tree building.

Example 4. *Continue the Examples 2 and 3. Calculate Deng entropy of BBAs of selected instance shown in Table 3:*

$$E^{SL}(m) = -0.7996 \times \log\frac{0.7996}{2^1-1} - 0.1757 \times \log\frac{0.1775}{2^2-1} - 0.0247 \times \frac{0.0247}{2^3-1} = 0.2933$$
$$E^{SW}(m) = -0.6904 \times \log\frac{0.6904}{2^1-1} - 0.2329 \times \log\frac{0.2329}{2^2-1} - 0.0767 \times \frac{0.0767}{2^3-1} = 0.3687$$
$$E^{PL}(m) = -1 \times \log\frac{1}{2^1-1} = 0$$
$$E^{PW}(m) = -1 \times \log\frac{1}{2^1-1} = 0$$

Similarly proceed same calculation to all instances so that average Deng entropy for attributes are calculated that

$$E\left(A^{SL}\right) = 1.3853,\ E\left(A^{SW}\right) = 2.0837,\ E\left(A^{PL}\right) = 0.4275,\ E\left(A^{PW}\right) = 0.2116.$$

According to this result, attribute PW will be chosen to generate child nodes.

Comparing the Deng entropy with the curves in Figure 2, we can intuitively learn that PW has the most distinctive curves for each class, yet curves in SW overlap each other a lot, which conforms to the size of the average Deng entropy above, where PW is the lowest and SW is the highest.

As a matter of fact, Examples 1–4 in this chapter can be orderly combined as a whole calculating example, which shows the procedure of the proposed attribute selecting method.

3.2. Splitting Strategy

The splitting strategy is redesigned according to the selected attribute A^* to fit the proposed attribute selection method. Branches will be associated to each class, that is to say, each node to be edged will have K branches. For an instance I_n, consider the generated BBAs, the class corresponding to the maximum mass value will be the branch that this instance shall be put into. To put it simply, when splitting the tree under attribute A^*, the instance I_n will be assigned into the k_n-*th* child node, where the k_n satisfies

$$k_n = \arg\max_k \phi^*_{nk}\left(x^d_n\right). \tag{34}$$

The Algorithm 2 summarizes the procedure of selecting attribute and splitting. It should be mentioned that, though the child nodes are associated to each class, this splitting strategy does not mean to determine the affiliation of instances directly and arbitrarily in this step.

Algorithm 2 Attribute selection and splitting.

Input: evidential training set $T_{pl} = (\boldsymbol{x}, pl_y)$, possible splitting attribute $\mathcal{A} = \{A^1, \ldots, A^D\}$
Output: selected attribute A^*, instance sets in child nodes $T_i, (i = 1, \ldots, K)$

compute the normal distribution parameters $\left(\mu_k^d, \sigma_k^d\right)$ for each A^d and C_k by E^2M algorithm;
for each attribute A^d **do**
  for each instance I^n **do**
    generate BBA m_n^d from normal distributions $\mathcal{N}_k^d\left(\mu_k^d, \sigma_k^{d2}\right), k = 1, \ldots, K$;
    $E_n^d = E\left(m_n^d\right)$; {Calculate Deng entropy for all generated BBAs}
  end for
  $E\left(A^d\right) = Average\left(E_n^d\right)$; {Calculate average Deng entropy for each attribute}
end for
split on attribute $A^* = \arg\min_{A^d}\left(E\left(A^d\right)\right)$;{The attribute with minimum average entropy is selected}

3.3. Stopping Criterion and Prediction Decision

After designing the attribute selection and partitioning strategy, we split each decision node to several child nodes. This procedure repeats iteratively until one of the stop criterion is met:

- No more attributes for selection;
- The number of instances in the nodes falls below a set threshold;
- The labels of instances are all precise and fall into the same class;

When the tree building stops at a leaf node L, a class label should be determined to predict the instances that fall into this node. We design two different prediction methods such that:

- The first one is to generate the prediction label from the original training labels of instances contained by this node, which is a similar treatment to traditional decision trees such as C4.5 tree method. Denoting the instances in the leaf node by $\{I_1', \ldots I_P'\}$ and the corresponding evidential training labels by $\left\{pl_p'\right\}, p = 1, \ldots P$, the leaf node will be labeled by $\hat{C}$, where

$$\hat{C} = \arg\max_{C_k} \sum_{p=1}^{P} pl_p'(C_k), k = 1, \ldots, K, \tag{35}$$

  which means the class label with maximal plausibility summation will represent this node. This tree predicts from the original labels of training set, which is called *Oringin-prediction belief entropy tree*(*OBE tree*) for short in this paper.
- The first method described above in fact abandons the generated BBAs during the tree build procedure, which will be adopted to generating predicted instance label in the second method. Firstly, the splitting attributes list, which lead instance I' to the leaf node from top to down, are denoted by $\left\{A_1^*, \ldots, A_Q^*\right\}$, and the BBAs generated accordingly are denoted by $m_1^* \ldots, m_Q^*$. Then combine these BBAs by Dempster rule, such that $\hat{m} = m_1^* \oplus \cdots \oplus m_Q^*$, to predict the training instance. On this basis, we continue to combine generated BBAs of all instances in a leaf node such that $\hat{m}_{leaf} = \hat{m}_1 \oplus \cdots \oplus \hat{m}_P$, where the once again combined BBA $\hat{m}_{leaf}$ will be the mass prediction label for the whole leaf node. To obtain a precise label for another choice, the last step is making decision on BBA by choosing the class label with maximal pignistic probability computed by Equation (11). We call this tree a *Leaf-prediction belief entropy tree*(*LBE tree*) in this paper.

The Algorithm 3 summarizes the induction of belief entropy trees introduced in this section.

Algorithm 3 Induction of belief entropy trees (BE-tree).

Input: evidential training set T_{pl}, classifier type $TYPE$
Output: belief entropy tree *Tree*

construct a root node containing all instances T_pl;
if stopping criterion is met **then**
  if $TYPE = OBE$ **then**
    output precise prediction generated from original plausibility label for the whole node;
  else if $TYPE = LBE$ **then**
    combine BBAs generates during each splitting $\hat{m} = m_1^* \oplus \cdots \oplus m_Q^*$ for each instance;
    combine BBAs of all instances in previous node generated in step 6 that $\hat{m}_{leaf} = \hat{m}_1 \oplus \cdots \oplus \hat{m}_P$;
    output $\hat{m}_{leaf}$ as a mass prediction for the whole leaf node;
    output $\hat{C} = Pignistic\left(\hat{m}_{leaf}\right)$ as a precise prediction for the whole leaf node;
  end if
  return *Tree*=root node;
else
  apply Algorithm 2 to select splitting attribute A^*;
  induce each subset $T_{pl_{child}}$ based on A^*;
  for all $T_{pl_{child}}$ **do**
    $Tree_{child} = BE\text{-}tree\left(T_{pl_{child}}\right)$;{Recursively build the tree on the new child node}
    attach $Tree_{child}$ to the corresponding *Tree*;
  end for
end if

3.4. An Alternative Method for Predicting New Instance

Two types of belief entropy trees, the OBE tree and the LBE tree, have been described in detail in the last section. Similar to traditional decision trees, a new instance will be classified in a top-down way: starting at the root node and following branches by considering its generated BBA under splitting attribute until reaching a leaf node. The prediction of leaf node will be given to this new instance.

However, differing from the idea of collecting the numerous 'opinions' of instances, another method to predict a new instance is considered after a tree has been built. In Section 3.1.2, we introduced how to generate each training instance's BBA corresponding to attributes. In the same way, we can generate $m_1^* \ldots, m_Q^*$ corresponding to an attributes list $\left\{A_1^*, \ldots, A_Q^*\right\}$, which orderly splits and leads the new instance to a leaf node. Then, we combine these BBAs such that $\hat{m} = m_1^* \oplus \cdots \oplus m_Q^*$ to predict the new testing instance. It is easy to find that this method performs the same way as the front part of label prediction in LBE trees, yet stops when obtaining a mass prediction from the testing instance's own attribution values instead of the leaf node it belongs to, which also means testing instances in a same leaf node normally have different mass prediction under this design. For the sake of narrative, a tree predicting in this way is called $Instance\text{-}prediction\ belief\ entropy\ tree (IBE\ tree)$ in this paper.

Figures 4 and 5 show the procedure of making prediction on leaf node, where Figure 4 is the generation of mass prediction $\hat{m}$ for each instance, in whether training set or testing set; Figure 5 details the different prediction making in the proposed three belief entropy trees.

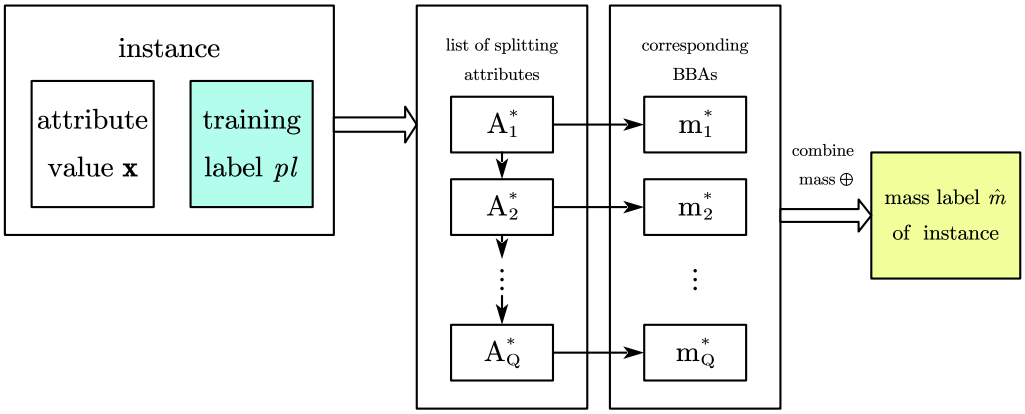


Figure 4. Generation of mass prediction for each instance.

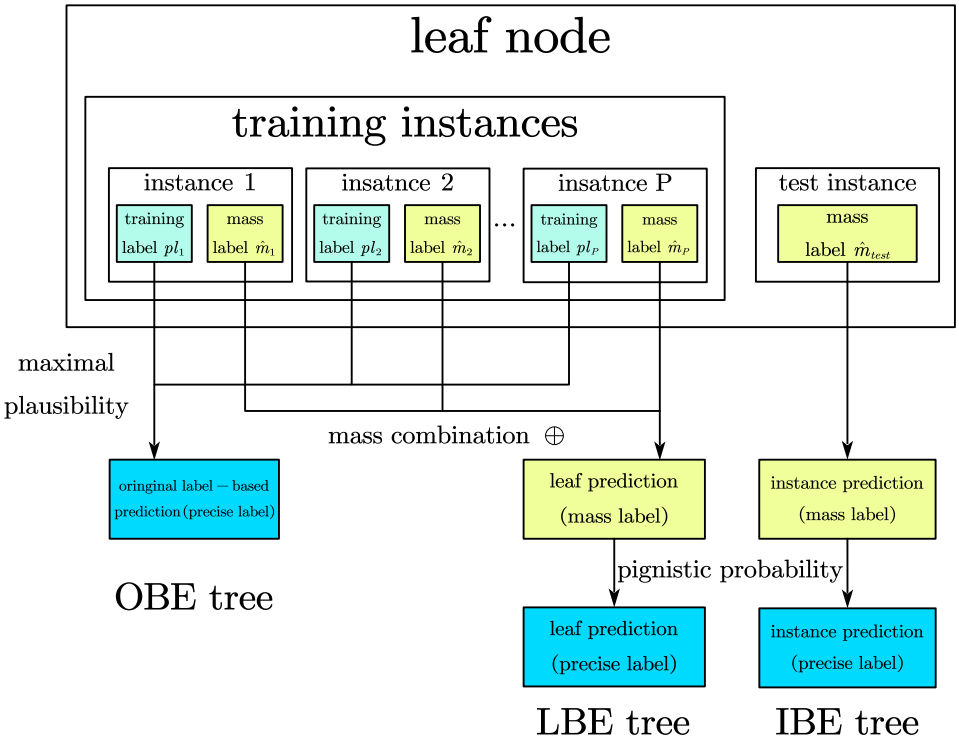


Figure 5. Three different ways to make prediction in belief entropy trees.

4. Belief Entropy Random Forest

We have introduced the induction of belief entropy trees in the previous section, which is regarded as the basic classifier of random forest ensemble method in the following discussion.

The generalization ability of random forest draws from not only the perturbation of sampling, but also the perturbation of attributes selecting. Specific to the proposed belief entropy random forest, for each basic tree, we firstly performs bootstrap sampling on the original training set, which means randomly sampling with replacement for N times on the set T where $|T| = N$. Secondly, when training on this resampling set, for each to-be-split node, the best splitting attribute will be chosen from a subset $\{A_i'\}, i = 1, \ldots, D'$ of the set of all available attributes $\{A_j\}, j = 1, \ldots, D$, where $1 < D' < D$. If $D' = D$, the basic tree splits totally, the same as the belief entropy tree; while $D' = 1$ means randomly selecting an attribute to split all the time.

Repeat the first and second steps above S times then a 'forest' containing variable basic trees will be constructed, where the repeat time S is called forest size. When making a

prediction of a new instance on this forest, S primary predictions will be independently generated by S basic trees and finally summarized to one result. It should be mentioned that the OBE tree output precise label directly for testing instances while the LBE trees and IBE trees can provide mass labels described by BBAs or precise labels. This feature inspires two different strategies for making predictions in the last step: the majority voting for precise labels and belief combination of mass labels.

Algorithm 4 shows the procedure of building the complete evidential random forests based on belief entropy trees. Selecting different ensemble prediction strategies and base tree types, we build five random forest lists below:

- label-voting OBE Random Forest(L-OBE RF), which performs majority voting on precise outputs of OBE trees;
- label-voting LBE Random Forest(L-LBE RF), which performs majority voting on precise outputs of LBE trees;
- mass-combination LBE Random Forest(M-LBE RF), which combines BBAs generated by LBE trees and makes decision;
- label-voting IBE Random Forest(L-IBE RF), which performs majority voting on precise outputs of IBE trees;
- mass-combination IBE Random Forest(M-IBE RF), which combines BBAs generated by IBE trees and makes decision;

Algorithm 4 Building procedure of evidential random forests.

Input: evidential training set T_{pl}, new instance $\boldsymbol{x}$, base classifier number h, base classifier type $TYPE$, base classifier output mode O
Output: predicted label $\hat{y}$

for $i = 1 : h$ **do**
  $T_i = RandomAttributeSampling\Big(RandomInstanceSampling\Big(T_{pl}\Big)\Big)$; {The resampling procedure of each base tree.}
  if $TYPE = OBE$ **then**
    $Tree_i = OBE(T_i)$;
  else if $TYPE = IBE$ **then**
    $Tree_i = IBE(T_i)$;
  else if $TYPE = LBE$ **then**
    $Tree_i = LBE(T_i)$;
  end if
end for
for $i = 1 : h$ **do**
  $L_i = Label\ Prediction(Tree_i, \boldsymbol{x})$; {Generate predict labels of each base tree.}
end for
if $O = precise\ label$ **then**
  $\hat{y} = Majority(L_1, \ldots, L_h)$; {Generate prediction from precise labels.}
else if $O = mass\ label$ **then**
  $\hat{y} = Pignistic(MassCombination(L_1, \ldots, L_h))$ {Generate prediction from mass labels.}
end if

Figure 6 shows the procedure of constructing the forests, in which the Figure 6a shows generation of basic trees in a random forest, and Figure 6b shows different procedure of combining the final prediction in five forests, which will lead to a different classification performance. We will evaluate them in the next section.

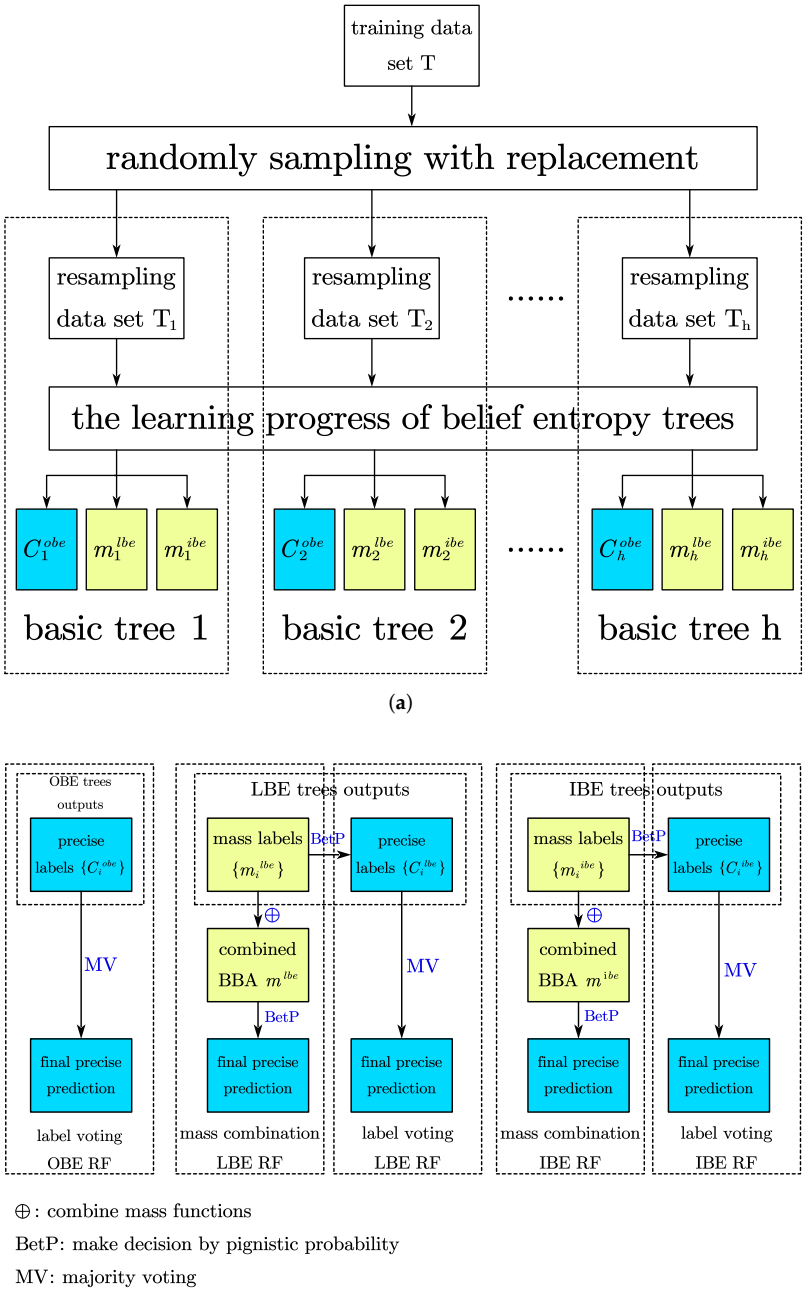


Figure 6. Belief entropy in random forests. (**a**) Generation of basic trees and their outputs in random forest. (**b**) Combination strategies of different random forests.

5. Experiments

In this section, we detail experiments to evaluate the performance of the proposed decision tree method. The experiment settings and results are detailed below.

5.1. Experiment Settings

As there are no widely accepted evidential data sets to measure the proposed method, it is necessary to generate a data set with ill-known labels from machine learning databases taken from the UCI repository [46]. We selected several data sets, including: Iris, Wine, Balance scale, Breast cancer, Sonar and Ionosphere.

Denote the true label of a instance by C_i, and give its uncertain observation m_{y_i}. Due to the characters of belief function, we can simulate several situations from precise data:

- a *precise* observation is such that $pl_{y_i}(C_i) = 1$, and $pl_{y_i}(C_j) = 0, \forall C_j \neq C_i^*$;
- a *vacuous* observation is such that $pl_{y_i}(C_j) = 1, \forall C_j \in \mathcal{C}$;
- an *imprecise* observation is such that $pl_{y_i}(C_j) = 1$ if $C_j = C_i^*$ or $C_j \in \mathcal{C}_{rm}$, and $pl_{y_i}(C_j) = 0$ otherwise, where $\mathcal{C}_{rm}$ is a set of randomly selected labels;
- an *uncertain* observation is such that $pl_{y_i}(C_i^*) = 1$, and $pl_{y_i}(C_j) = r_j, \forall C_j \neq C_i^*$, where r_j are sampled independently from uniform distribution $\mathcal{U}([0,1])$.

To observe the performance on evidential training data sets with different ill-known types and incomplete degrees, we set three variables, vacuousness level $V \in [0,1]$, imprecision level $I \in [0,1]$ and uncertainty level $U \in [0,1]$, to adjust the generation procedure, where $V + I + U \leqslant 1$.

Example 2 shows the transformed Iris data set and listed part of instances in Table 1. In this example, labels of no.53 and no.54 instance are vacuous; labels of no.1 and no.2 instance are imprecise; labels of no.4 and no.52 instance are uncertain.

To improve the reliability and reduce the stochasticity, we performed 5-fold cross-validation on each data set and repeat ten times to compute an average classification accuracy for all experiments. Different tree induction techniques will be compared:

- *traditional C*4.5 *tree*, which only uses precise data in the training set during tree induction;
- *belief entropy trees* described in Section 3.3: OBE tree, LBE tree, IBE tree;
- *belief entropy random forest* described in Section 4:
  - *label-voting OBE random forest*(*L-OBE RF*);
  - *label-voting LBE random forest*(*L-LBE RF*);
  - *mass-combination LBE random forest*(*M-LBE RF*);
  - *label-voting IBE random forest*(*L-IBE RF*);
  - *mass-combination IBE random forest*(*M-IBE RF*);

We set the maximal size of the leaf node as $|T|/20$ to avoid overfitting in the belief entropy trees. In the random forests, the forest size was set as 50, and the size of the attributes subset was set as $D' = log_2 D$.

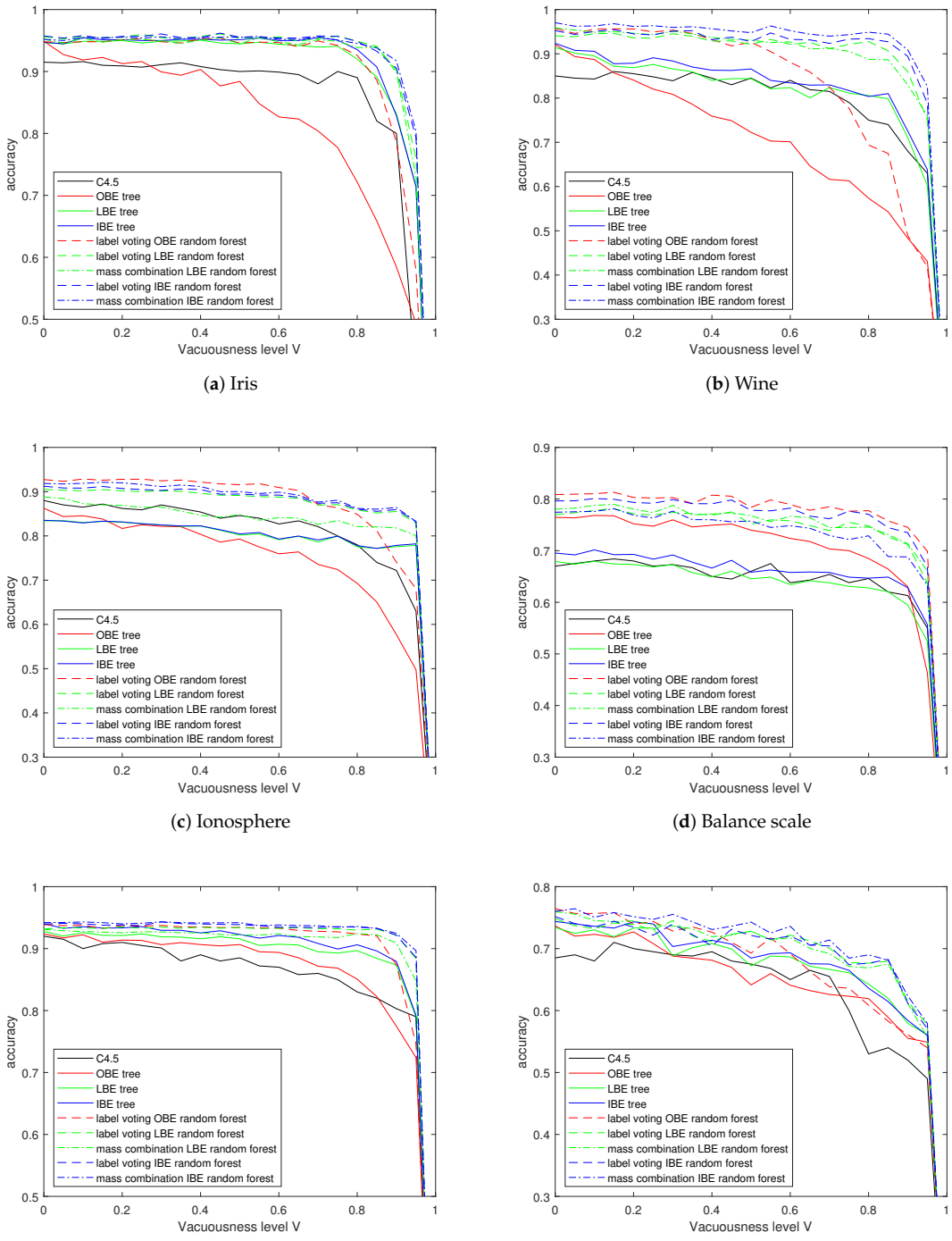
5.2. Experiments on Vacuous Data

Assuming part of the instances in the training set are totally unobserved while others are completely observed, we performed experiments with different vacuousness levels $V \in [0,1]$ while $I = U = 0$. Generating the training sets and learning on them, the results are shown in Figure 7.

Firstly, we observe the figure as a whole. Obviously, whatever the tree induction method is, it is impossible to learn from data sets whose instances are all vacuous. Thus, the accuracy of all trees decreases gradually as V increases, yet drops sharply when the V approaches nearly to 1. On the contrary, almost all curves keep steady or decrease slightly before the vacuousness level reaches 80%, except for the OBE trees. Table 4 shows the accuracy results when V equals 90%.

Table 4. Classification accuracy on UCI data sets with 90% vacuousness level.

	Iris	Wine	Ionosphere	Balance Scale	Breast Cancer	Sonar
C4.5	0.800	0.6814	0.7221	0.6134	0.8031	0.5247
OBE tree	0.5846	0.4038	0.5775	0.6305	0.7751	0.5553
LBE tree	0.8326	0.6026	0.7763	0.5954	0.8732	0.5793
IBE tree	0.8288	0.6386	0.7797	0.6282	0.8793	0.5841
label voting OBE Random Forest	0.7864	0.4897	0.7404	0.7453	0.8745	0.5611
label voting LBE Random Forest	0.9020	0.8594	0.8555	0.7127	0.9319	0.6149
mass combination LBE Random Forest	0.8989	0.8295	0.8182	0.7138	0.9223	0.6115
label voting IBE Random Forest	0.9053	0.8940	0.8608	**0.7454**	**0.9338**	0.6120
mass combination IBE Random Forest	**0.9174**	**0.9082**	**0.8647**	0.6891	0.9330	**0.6236**



(**a**) Iris (**b**) Wine

(**c**) Ionosphere (**d**) Balance scale

(**e**) Breast cancer (**f**) Sonar

Figure 7. Classification accuracy on UCI data sets with different vacuousness levels.

Considering the basic belief entropy trees firstly, the LBE trees and IBE trees perform, most of the time, at least as well as the traditional C4.5 decision trees, and better than the traditional decision trees for some time, especially when encountering high vacuousness level V; however, the OBE preforms elusively on different data sets: it has the lowest classification accuracy in Iris, Wine and Ionosphere data set; however, it achieves better results in the Balance scale. It is possible that if all samples in a leaf node are vacuous, the direct combination of all the training labels stays vacuous, which led to the shortage of OBE tree.

It can be observed that the belief entropy random forests perform well overall for their improvement in classification accuracy compared to the corresponding basic tree and the slower accuracy decent rate as V increases. Among these forests, the ones based on IBE and making prediction by mass combination performs better than others in nearly all data sets except the Balance scale.

5.3. Experiments on Imprecise Data

The second situation is that some data are imprecisely observed, i.e., the observation is a set value, while the true value lies in this set (called superset labels [47] in some works). As mentioned before, imprecision level I controls the percentage of imprecise observations.

For the instance to be imprecise, we randomly generate a number $z_k \in [0, 1]$ for each class C_k except the true one. Plausibility of labels with $z_k < I$ will be set to 1. When the $I = 1$, a training set becomes totally imprecise, which is, in practice, the same situation as total vacuousness; while $I < 1$, instances are in a middle state of transition from precise to vacuous, which indicates a piece of similarity between the vacuous training set and the imprecise training set, i.e., we can tell that the imprecise sample contains more information than the totally vacuous ones. As a result, we can see in Figure 8, that curves of accuracy with changing I are similar to those in experiments with vacuousness in Figure 7, yet more smooth and full.

According to the Table 5, the proposed methods keep pretty good classification results under high-level imprecise observations. OBE still keeps the shortage in almost all data sets while LBE and IBE achieve similar performance. *M-IBE RF* keeps its advantage in most situations, especially in the Iris and Breast Cancer data; the classification accuracy is almost equal to the results on the total precise training set. The balance scale is a particular case to be discussed later.

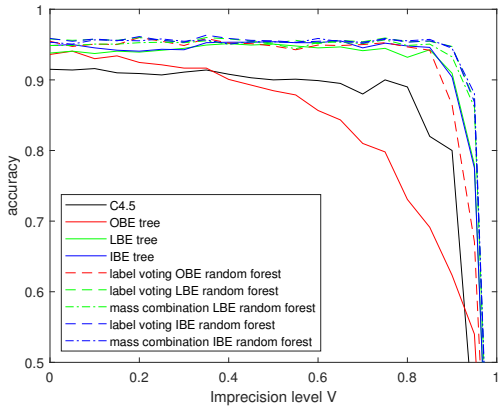
Table 5. Classification accuracy on UCI data sets with 90% imprecision level.

	Iris	**Wine**	**Ionosphere**	**Balance Scale**	**Breast Cancer**	**Sonar**
C4.5	0.8000	0.6814	0.7221	0.6134	0.8031	0.5247
OBE tree	0.6233	0.5382	0.6786	0.6746	0.8605	0.5841
LBE tree	0.9093	0.7719	0.7858	0.6146	0.8979	0.6303
IBE tree	0.9040	0.7899	0.7892	0.6381	0.9051	0.6413
label voting OBE random forest	0.8647	0.6208	0.8552	0.7536	0.9257	0.6351
label voting LBE random forest	0.9473	0.9124	0.8621	0.7483	0.9359	0.6630
mass combination LBE random forest	0.9327	0.9118	0.8259	0.7437	0.9262	0.6635
label voting IBE random forest	**0.9467**	0.9219	0.8684	**0.7709**	0.9364	0.6572
mass combination IBE random forest	0.9447	**0.9326**	**0.8755**	0.7237	**0.9399**	**0.6702**

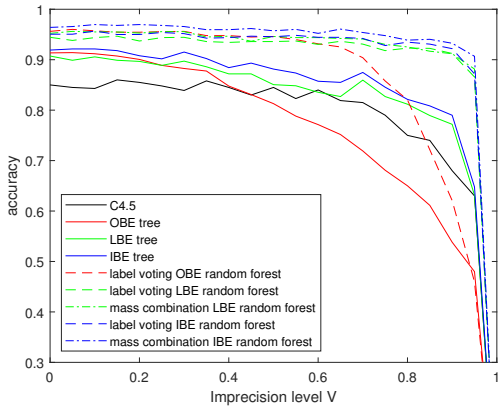
5.4. Experiments on Uncertain Data

Another type of ill-known label is the uncertain one, which is measured by a plausibility distribution, with the true label having the highest chance among all class labels. To evaluate the performance of the proposed trees and forests in a more general situation with uncertainty, we set $U \in [0, 1]$ and $V = I = 0$. For instance, to be transformed into an uncertain one, we assign a value 1 to the plausibility of the true label and random values averagely sampled from $[0, U]$ to other labels.

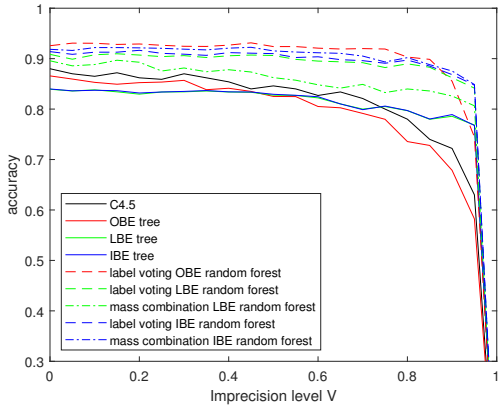
Despite the inability to handle total vacuousness and imprecision, the belief entropy trees have the ability to learn from totally uncertain training data sets. The horizontal curves in Figure 9 indicate all methods proposed in this paper keep stable performance with changing uncertainty level U. On the whole, we can learn from the figure that LBE and IBE perform equally well and better than OBE as a single tree in most data sets, except in the Balance scale.



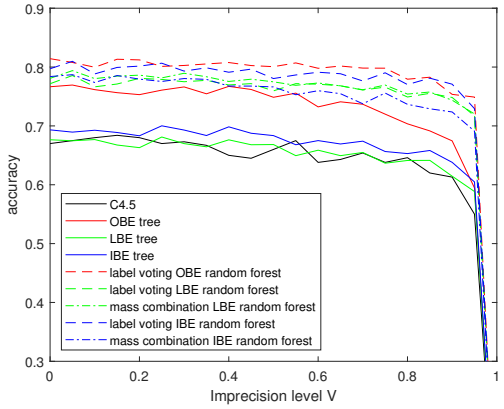
(**a**) Iris



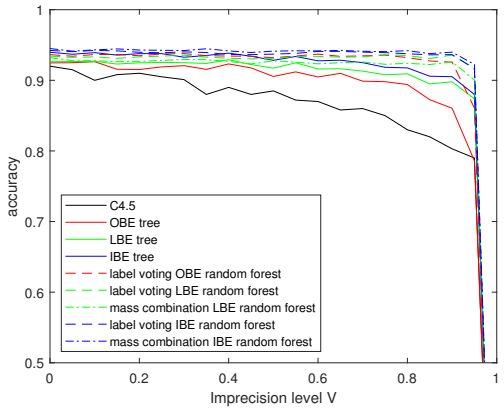
(**b**) Wine



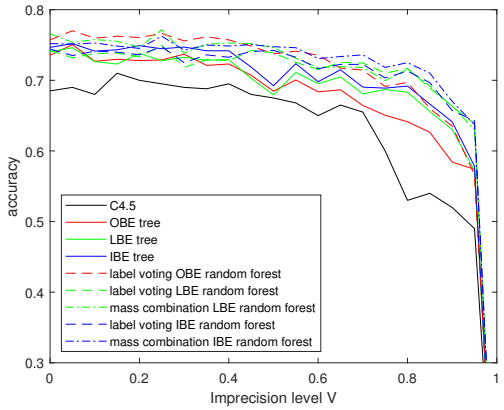
(**c**) Ionosphere



(**d**) Balance scale

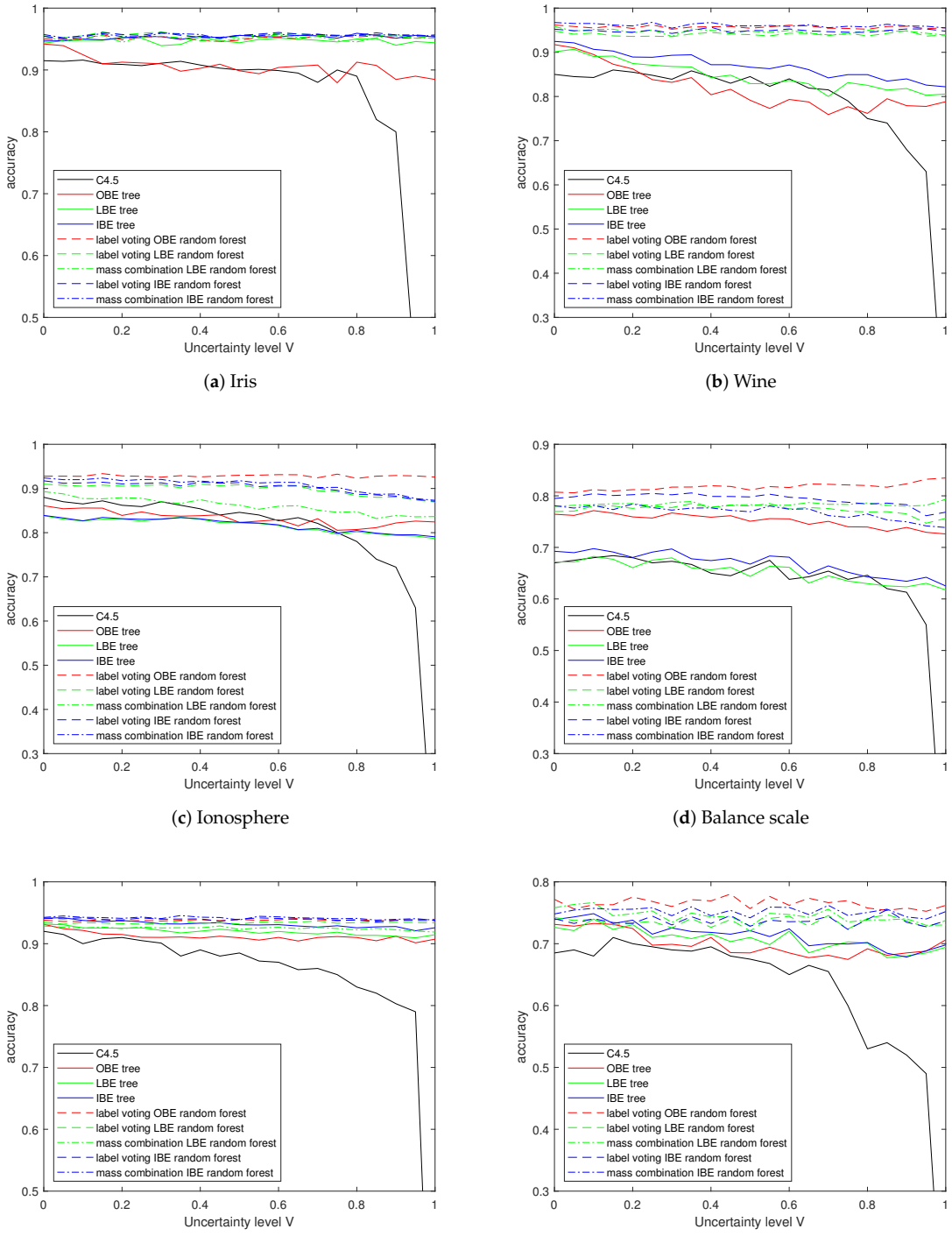


(**e**) Breast cancer



(**f**) Sonar

Figure 8. Classification accuracy on UCI data sets with different imprecision levels.



(**a**) Iris (**b**) Wine

(**c**) Ionosphere (**d**) Balance scale

(**e**) Breast cancer (**f**) Sonar

Figure 9. Classification accuracy on UCI data sets with different uncertainty levels.

Considering the forests, for the good attribute normality of Iris, Wine and Breast cancer data, classification accuracies of the five forests on these data sets have similar performance according to Table 6, leading to a heavy overlap of curves in figure. Among these trees, the OBE trees achieve the most significant improvement by building random forest; this improvement helps OBE-RF to surpass other forests in the Ionosphere, Sonar and Balance scale data sets. Particularly, in the Balance scale, the accuracy of OBE-RF even increases slightly as the U decreases, which can be partially explained by the fact that uncertain instances are more informative then absolutely precise instances.

Table 6. Classification accuracy on UCI data sets with 90% uncertainty level.

	Iris	Wine	Ionosphere	Balance Scale	Breast Cancer	Sonar
C4.5	0.8000	0.6814	0.7221	0.6134	0.8031	0.5247
OBE tree	0.8847	0.7792	0.8219	0.7387	0.9121	0.6851
LBE tree	0.9400	0.8180	0.7937	0.6235	0.9120	0.6803
IBE tree	0.9513	0.8399	0.7955	0.6342	0.9278	0.6784
label voting OBE random forest	0.9513	0.9590	**0.9293**	**0.8233**	0.9381	**0.7577**
label voting LBE random forest	0.9547	0.9489	0.8803	0.7651	0.9344	0.7370
mass combination LBE random forest	0.9560	0.9478	0.8390	0.7800	0.9225	0.7404
label voting IBE random forest	**0.9567**	0.9528	0.8826	0.7829	0.9376	0.7346
mass combination IBE random forest	**0.9567**	**0.9596**	0.8875	0.7496	**0.9394**	0..7433

5.5. Summary

By carrying out experiments on training sets with different types and degrees of incomplete observation, we can conclude that the LBE trees and IBE trees, along with four types of random forests based on them, generally possess excellent learning ability on data with ill-known labels. Among the RFs, the ensemble of the IBE tree, L-IBE-RF and M-IBE RF achieve the highest classification accuracy in most situations except on samples with high uncertainty levels, especially on the Balance scale data set. We think there are two reasons: (a) compared to vacuous and imprecise samples, the learning labels of uncertain samples are more information rich, while the OBE use the learning labels to predict directly; (b) the attribute values of Ionosphere, Balance, and Sonar data sets contain less normality than others—the balance scale are totally not normal. We can conclude that the ensemble OBE RF requests less normality of the data set.

The results of experiments indicate that the application of the belief function tool to the prediction of trees and combination of forests is efficient and reasonable; yet there are also some drawbacks. Firstly, the introduction of the belief function and mass combination obviously increases the time cost of learning. The sensitivity to the normality of data makes the proposed trees and RFs unable to handle, to the greatest extent, all situations with one particular structure.

6. Conclusions

In this paper, a new classification tree method based on belief entropy is proposed to cope with uncertain data. This method directly models continuous attribute values of training data by E^2M algorithm, and selects a splitting attribute via a new tool–belief entropy. Differing from the traditional decision trees, we redesign the splitting and prediction, making them fit the feature of uncertain labels described by the belief function. Finally, random forests with different combination strategies were constructed on the basis of the proposed tree method to seek higher accuracy and stronger generalization ability.

As the experimental results show, the proposed belief entropy trees are robust to different sorts of uncertainty. They perform closely to traditional decision trees on precise data and keep good results on data with ill-known labels. Meanwhile, the belief entropy random forests, which improve significantly when compared to the basic belief function trees, achieve excellent and stable performance even in the situation with high-level uncertainty. It is proved that the proposed trees and random forests have a potentially broad field of application. In future research, some further improvements will be investigated, such as more reasonable BBA combination methods for the incapacity of Dempster's rule to handle huge mass conflict, and a boosting ensemble method based on the belief entropy trees.

Author Contributions: Conceptualization, K.G. and Y.W.; methodology, K.G. and L.M.; software, K.G.; validation, K.G. and L.M.; formal analysis, K.G.; investigation, K.G. and L.M.; resources, Y.W.; data curation, K.G.; writing—original draft preparation, K.G.; writing—review and editing, L.M. and Y.W.; supervision, Y.W.; funding acquisition, Y.W. All authors have read and agreed to the published version of the manuscript.

Funding: The work described in this paper was supported by the National Natural Science Foundation of China (61973291).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors greatly appreciate the reviewers' suggestions and the editor's encouragement.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Josse, J.; Chavent, M.; Liquet, B.; Husson, F. Handling missing values with regularized iterative multiple correspondence analysis. *J. Classif.* **2012**, *29*, 91–116. [CrossRef]
2. Quinlan, J.R. Decision trees as probabilistic classifiers. In Proceedings of the Fourth International Workshop on Machine Learning, Irvine, CA, USA, 22–25 June 1987; Elsevier: Amsterdam, The Netherlands, 1987; pp. 31–37.
3. Tsang, S.; Kao, B.; Yip, K.Y.; Ho, W.S.; Lee, S.D. Decision trees for uncertain data. *IEEE Trans. Knowl. Data Eng.* **2009**, *23*, 64–78. [CrossRef]
4. Couso Blanco, I.; Sánchez Ramos, L. Harnessing the information contained in low-quality data sources. *Int. J. Approx. Reason.* **2014**, 1485–1486. http://dx.doi.org/10.1016/j.ijar.2014.05.006. [CrossRef]
5. Masson, M.H.; Denoeux, T. Ranking from pairwise comparisons in the belief functions framework. In *Belief Functions: Theory and Applications*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 311–318.
6. Yuan, Y.; Shaw, M.J. Induction of fuzzy decision trees. *Fuzzy Sets Syst.* **1995**, *69*, 125–139. [CrossRef]
7. Wang, X.; Liu, X.; Pedrycz, W.; Zhang, L. Fuzzy rule based decision trees. *Pattern Recognit.* **2015**, *48*, 50–59. [CrossRef]
8. Hüllermeier, E. Possibilistic induction in decision-tree learning. In *Lecture Notes in Computer Science : Proceedings of the European Conference on Machine Learning*; Springer: Berlin/Heidelberg, Germany, 2002; pp. 173–184.
9. Qin, B.; Xia, Y.; Li, F. DTU: A decision tree for uncertain data. In *Lecture Notes in Computer Science: Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 4–15.
10. Liang, C.; Zhang, Y.; Song, Q. Decision tree for dynamic and uncertain data streams. In Proceedings of the 2nd Asian Conference on Machine Learning. JMLR Workshop and Conference Proceedings, Tokyo, Japan, 8–10 November 2010; pp. 209–224.
11. Dempster, A.P. Upper and lower probabilities induced by a multivalued mapping. In *Classic Works of the Dempster-Shafer Theory of Belief Functions*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 57–72.
12. Shafer, G. *A Mathematical Theory of Eevidence*; Princeton University Press: Princeton, NJ, USA, 1976.
13. Elouedi, Z.; Mellouli, K.; Smets, P. Belief decision trees: Theoretical foundations. *Int. J. Approx. Reason.* **2001**, *28*, 91–124. [CrossRef]
14. Trabelsi, S.; Elouedi, Z.; Mellouli, K. Pruning belief decision tree methods in averaging and conjunctive approaches. *Int. J. Approx. Reason.* **2007**, *46*, 568–595. [CrossRef]
15. Vannoorenberghe, P.; Denoeux, T. Handling uncertain labels in multiclass problems using belief decision trees. In Proceedings of the IPMU, Annecy, France, 1–5 July 2002; Volume 3, pp. 1919–1926.
16. Sutton-Charani, N.; Destercke, S.; Denœux, T. Classification trees based on belief functions. In *Belief Functions: Theory and Applications*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 77–84.
17. Sutton-Charani, N.; Destercke, S.; Denœux, T. Learning decision trees from uncertain data with an evidential EM approach. In Proceedings of the 12th International Conference on Machine Learning and Applications, Miami, FL, USA, 4–7 December 2013; Volume 1, pp. 111–116.
18. Sutton-Charani, N.; Destercke, S.; Denœux, T. Training and evaluating classifiers from evidential data: Application to E^2M decision tree pruning. In *Lecture Notes in Computer Science: Proceedings of the International Conference on Belief Functions*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 87–94.
19. Ma, L.; Destercke, S.; Wang, Y. Online active learning of decision trees with evidential data. *Pattern Recognit.* **2016**, *52*, 33–45. [CrossRef]
20. Trabelsi, A.; Elouedi, Z.; Lefevre, E. Handling uncertain attribute values in decision tree classifier using the belief function theory. In *Lecture Notes in Computer Science: Proceedings of the International Conference on Artificial Intelligence: Methodology, Systems, and Aapplications*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 26–35.

21. Trabelsi, A.; Elouedi, Z.; Lefevre, E. New decision tree classifier for dealing with partially uncertain data. In Proceedings of the 25th Rencontres Francophones sur la Logique Floue et ses Applications (LFA 2016), Nimes, France, 23 November 2016; pp. 57–64.
22. Trabelsi, A.; Elouedi, Z.; Lefevre, E. Decision tree classifiers for evidential attribute values and class labels. *Fuzzy Sets Syst.* **2019**, *366*, 46–62. [CrossRef]
23. Kim, B.; Jeong, Y.S.; Tong, S.H.; Jeong, M.K. A generalised uncertain decision tree for defect classification of multiple wafer maps. *Int. J. Prod. Res.* **2020**, *58*, 2805–2821. [CrossRef]
24. Zou, J.; Yan, X.; Zhou, Y. Discounted Belief Decision Tree for Uncertainty data from unreliable source. In Proceedings of the 2019 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 5–7 December 2019; pp. 382–387.
25. Li, Y.; Chen, W. Landslide susceptibility evaluation using hybrid integration of evidential belief function and machine learning techniques. *Water* **2020**, *12*, 113. [CrossRef]
26. Denœux, T. Maximum likelihood from evidential data: An extension of the EM algorithm. In *Combining Soft Computing and Statistical Methods in Data Analysis*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 181–188.
27. Denoeux, T. Maximum likelihood estimation from uncertain data in the belief function framework. *IEEE Trans. Knowl. Data Eng.* **2011**, *25*, 119–130. [CrossRef]
28. Deng, Y. Deng entropy. *Chaos Solitons Fractals* **2016**, *91*, 549–553. [CrossRef]
29. Ma, L.; Sun, B.; Han, C. Training instance random sampling based evidential classification forest algorithms. In Proceedings of the 2018 21st International Conference on Information Fusion (FUSION), Cambridge, UK, 10–13 July 2018; pp. 883–888.
30. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
31. Gao, K.; Ma, L.; Wang, Y. A Classification Tree Method Based on Belief Entropy for Evidential Data. In *Lecture Notes in Computer Science: Proceedings of the International Conference on Belief Functions*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 105–114.
32. Quinlan, J.R. Induction of decision trees. *Mach. Learn.* **1986**, *1*, 81–106. [CrossRef]
33. Quinlan, J.R. *C4.5: Programs for Machine Learning*; Elsevier: Amsterdam, The Netherlands, 2014.
34. Breiman, L.; Friedman, J.H.; Olshen, R.A.; Stone, C.J. *Classification and Regression Trees*; Routledge: London, UK, 2017.
35. Ma, L.; Denoeux, T. Partial classification in the belief function framework. *Knowl.-Based Syst.* **2021**, *214*, 106742. [CrossRef]
36. Denoeux, T. Decision-making with belief functions: A review. *Int. J. Approx. Reason.* **2019**, *109*, 87–110. [CrossRef]
37. Smets, P. Decision making in the TBM: The necessity of the pignistic transformation. *Int. J. Approx. Rreasoning* **2005**, *38*, 133–147. [CrossRef]
38. Denoeux, T. Likelihood-based belief function: Justification and some extensions to low-quality data. *Int. J. Approx. Reason.* **2014**, *55*, 1535–1547. [CrossRef]
39. Dempster, A.P.; Laird, N.M.; Rubin, D.B. Maximum likelihood from incomplete data via the EM algorithm. *J. R. Stat. Soc. Ser. B (Methodol.)* **1977**, *39*, 1–22.
40. Shannon, C.E. A mathematical theory of communication. *ACM Sigmobile Mob. Comput. Commun. Rev.* **2001**, *5*, 3–55. [CrossRef]
41. Xu, P.; Deng, Y.; Su, X.; Mahadevan, S. A new method to determine basic probability assignment from training data. *Knowl.-Based Syst.* **2013**, *46*, 69–80. [CrossRef]
42. Li, M.; Xu, H.; Deng, Y. Evidential decision tree based on belief entropy. *Entropy* **2019**, *21*, 897. [CrossRef]
43. Fisher, R.A. The use of multiple measurements in taxonomic problems. *Ann. Eeugenics* **1936**, *7*, 179–188. [CrossRef]
44. McLachlan, G.J.; Peel, D. *Finite Mixture Models*; John Wiley & Sons: Hoboken, NJ, USA, 2004.
45. Quost, B.; Denoeux, T.; Li, S. Parametric classification with soft labels using the evidential EM algorithm: Linear discriminant analysis versus logistic regression. *Adv. Data Anal. Classif.* **2017**, *11*, 659–690. [CrossRef]
46. Dua, D.; Graff, C. UCI Machine Learning Repository. 2017. Available online: https://ergodicity.net/2013/07/ (accessed on 25 March 2022).
47. Liu, L.; Dietterich, T.G. A conditional multinomial mixture model for superset label learning. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2012; pp. 548–556.