



Ensemble of randomized soft decision trees for robust classification

G KISHOR KUMAR^{1,*}, P VISWANATH² and A ANANDA RAO³

¹Department of Information Technology, Rajeev Gandhi Memorial College of Engineering and Technology, Nandyal 518501, Andhra Pradesh, India

²Department of Computer Science and Engineering, Indian Institute of Information Technology Chittoor, Sri City 517541, Andhra Pradesh, India

³Department of Computer Science and Engineering, Jawaharlal Nehru Technological University Anantapur College of Engineering, Anantapuramu 515002, Andhra Pradesh, India
e-mail: kishorgulla@yahoo.co.in; viswanath.p@ieee.org; akepogu@gmail.com

MS received 10 March 2014; revised 23 May 2015; accepted 26 December 2015

Abstract. For classification, *decision trees* have become very popular because of its simplicity, interpret-ability and good performance. To induce a decision tree classifier for data having continuous valued attributes, the most common approach is, split the continuous attribute range into a hard (crisp) partition having two or more blocks, using one or several crisp (sharp) cut points. But, this can make the resulting decision tree, very sensitive to noise. An existing solution to this problem is to split the continuous attribute into a fuzzy partition (soft partition) using soft or fuzzy cut points which is based on fuzzy set theory and to use fuzzy decisions at nodes of the tree. These are called *soft decision trees* in the literature which are shown to perform better than conventional decision trees, especially in the presence of noise. Current paper, first proposes to use an *ensemble of soft decision trees* for robust classification where the attribute, fuzzy cut point, etc. parameters are chosen randomly from a probability distribution of fuzzy information gain for various attributes and for their various cut points. Further, the paper proposes to use probability based information gain to achieve better results. The effectiveness of the proposed method is shown by experimental studies carried out using three standard data sets. It is found that an ensemble of randomized soft decision trees has outperformed the related existing soft decision tree. Robustness against the presence of noise is shown by injecting various levels of noise into the training set and a comparison is drawn with other related methods which favors the proposed method.

Keywords. Decision tree; soft decision tree; bootstrapping; ensemble technique; fuzzy set.

1. Introduction

Data Mining is a process which discovers knowledge from large volumes of data by applying data analysis and discovery algorithms [1, 2]. Classification, a major data mining functionality, is a supervised learning method where the example set called the training set is used to classify the given query data item into one of the predefined classes, where a classifier derived from the training set like a decision tree or a neural network or a support vector machine or any other classifier may be used [3].

No Free Lunch [3] theorem states that there is no the best classifier suitable for all problems. In rare cases where the probability structure of the problem is fully known, the *Bayes classifier* gives the minimum error and hence is the best one. However, one has to, often work with a limited training set, even to derive probabilities like class conditional densities.

So, the Bayes classifier which uses estimated probabilities (from a limited training set) is not the best one. Some recent classifiers like support vector machines [4, 5] are shown to have good generalization ability especially with limited training set problems. But these methods and many other methods like artificial neural networks have an important limitation called lack of interpret-ability along with its inability to work with categorical or non-numeric attributes. That is, they cannot give reasons (in human understandable terms) for the decision it made. Instance based classifiers like nearest neighbor classifier [6] or case based methods [7] show good performance and also has good interpret-ability and can deal with non-numeric attributes provided a proper distance measure or rules are available. However the classification time these classifiers take is often proportionate to the training set (or rule set) size, hence are slow. Decision tree classifiers, where the classification rules are arranged in the form of a tree show good performance, are applicable to work with non-numeric attributes and also give good interpretations. Because of the tree structure, they are faster

*For correspondence

than other rule-based methods [8, 9]. Hence, decision tree classifiers are popular in data mining applications.

In the decision tree induction process, for a node that corresponds to a continuous valued attribute, the testing is done in two standard ways, *viz.*, by splitting the attribute range into two intervals using a cut point [10], or into many intervals using several cut points [11]. That is, a partition of the range can be found, which can be called *hard* discretization of the range of values. Conventional decision trees which are built by using continuous valued attributes, with a crisp threshold based testing at a node, do not perform well, especially in the presence of noise [12, 13]. Two objects which are very near in the feature space can be classified into different classes. Hence, the crisp threshold (cut point) is not always desirable [14], and it may produce high misclassification rates since it has poor generalization ability. This is one of the reasons why decision trees are unstable, *i.e.*, a small perturbation in the training set can result in a drastically different decision tree. This is the reason why decision trees have high variance in their error component [15]. To overcome this problem, some techniques use probability based approaches [16–18], some use possibility based methods [19–21]. One of the approaches to solve this problem is to use fuzzy¹ decision trees where standard decision trees are combined with fuzzy rules.

An improvement over the above one is to derive a soft partition of the range called *soft* discretization [21, 22] using fuzzy-set theory. Fuzzy decision trees have been used for many applications like diagnosis systems [23], video mining [24], landslide susceptibility detection [25] and to many other classification problems. Moreover, fuzzy decision trees have good interpretability than standard decision trees. Decision trees, in classifying a test pattern, use a single rule that corresponds to the path from the root to a leaf. All other rules are not satisfied. However, several rules might be satisfied with varying degrees of precision, which can be combined at a later stage to get the class label. This approach can reduce the variance component of the error [15]. A fuzzy decision tree induction method to solve regression problems was introduced in [15] where growing and pruning are combined to determine the structure of the soft decision tree with the techniques called refitting and back-fitting. The effect of using various information measures in building fuzzy decision trees is studied in [19].

An ensemble of classifiers, rather than using a single one is also known to reduce variance component of the error [26–28]. The most popular techniques for constructing ensembles are, (i) bootstrap aggregation or bagging [29, 30], (ii) the Adaboost family of algorithms [31] (called *boosting*) and (iii) random forests [32, 33]. Randomness was used in the induction process of decision trees, which are shown to result in a better classifier [34]. Random forests based decision trees perform significantly better than bagging and

boosting [35]. These decision trees have smaller error rate and are also more robust to noise than the other methods [35]. In [36] a completely random decision tree algorithm is proposed, where at each node the splitting attribute is randomly chosen without using any heuristic measures such as information gain, gain ratio and gini index. It means the choice of selection of attributes at each node is completely stochastic. Finally, the authors have concluded using experimental results that their model achieves significantly higher accuracy than the single hypothesis. In [37] the authors have clearly justified about the reason for higher accuracy of randomized decision tree approaches using posterior probability estimations. The authors also show that the mean squared error (MSE) criteria by randomized decision trees is reduced in both bias and variance. The other method is [38] where a subset of features is randomly generated from the original feature set and on each subset a conventional decision tree is induced and shown that their computations are better. Dietterich's method [39], where the splitting attribute is randomly chosen among the top k attributes which have highest information gain. In recent works, ensemble methods are combined with fuzzy-set based learning algorithms [40–43]. Another recent work based on ensemble method is a fuzzy random forest, where randomization is used to select a splitting attribute from a set of candidate attributes at every node [44]. A comparative study of combination methods of fuzzy Vs non-fuzzy based classifiers was done and concluded that better results are obtained with fuzzy based classifiers [45]. A good theoretical analysis was shown in [46] about how linear combiners of classifiers perform well, which depends on the performance of individual classifiers.

This paper, basically extends the ideas presented in [10] and [21] where a soft partition of the predefined width (informally, this will measure the degree of overlap between blocks of the soft partition) is obtained for continuous valued attributes. Since there is an inherent vagueness in choosing the attribute, its cut point, its width, etc., the paper proposes to build an ensemble of soft decision trees where the parameters (like the attribute, cut point, etc.) are chosen from probability distributions (which are obtained from the training set). Because of the randomness injected at various levels, the component classifiers are supposed to be sufficiently independent of each other, so that their consensus decision is going to be a better one.

The rest of the paper is organized as follows. Section 2 discusses conventional decision trees along with their shortcomings. Section 3 initially gives few definitions needed to describe fuzzy systems, then, presents an induction method of the basic soft decision tree. Section 4 describes the core of the paper, *i.e.*, a randomized soft decision tree induction method, followed by the discussion of an ensemble of randomized soft decision trees. Experimental studies are given in Section 6. Section 7 concludes the paper.

¹Fuzzy and soft, these two words are interchangeably used in this paper.

2. Standard decision tree induction

Let the data set be $\mathcal{X} = \{(X_i, y_i) \mid i = 1, 2, \dots, n\}$, where X_i represents a d -dimensional feature vector and y_i represents the corresponding class label. Let the set of attributes (features) be $\mathcal{A} = \{A_1, A_2, \dots, A_j, \dots, A_d\}$, which are all assumed to be of continuous valued.²

To insert a node in the decision tree building process, the goodness of the attribute to be tested at the node is normally found based on drop in certain impurity measure, like entropy, gini-index, and gain-ratio [1]. Usually entropy based impurity measure called *Information Gain* is used for continuous valued attributes which is described below. Let the data set be \mathcal{X} , its entropy is given by

$$Entropy(\mathcal{X}) = \sum_{i=1}^c -p_i \log_2(p_i), \quad (1)$$

where c represents number of class labels, for our experiments c is 2, p_i represents the probability of data instances that belong to class i .

Let $a_{i_1}, a_{i_2}, \dots, a_{i_m}$ be an ordered sequence of distinct values of an attribute A_i , $i \in \{1, 2, \dots, d\}$, as given in the training set. Then for every pair of values $(a_{i_k}, a_{i_{k+1}})$ a possible threshold $\tau_k = (a_{i_k} + a_{i_{k+1}})/2$, for $k = 1$ to $i_m - 1$, is examined for a cut point to partition \mathcal{X} into two blocks $\mathcal{X}_1^{(i)}$ and $\mathcal{X}_2^{(i)}$. Weighted average of the entropy of the resulting class distributions after the testing would be

$$Info(A_i, \tau_k) = \frac{n_1}{n} Entropy(\mathcal{X}_1^{(i)}) + \frac{n_2}{n} Entropy(\mathcal{X}_2^{(i)}), \quad (2)$$

where n_1 and n_2 are the number of tuples in blocks $\mathcal{X}_1^{(i)}$ and $\mathcal{X}_2^{(i)}$. Information Gain G of this testing is

$$G(A_i, \tau_k) = Entropy(\mathcal{X}) - Info(A_i, \tau_k). \quad (3)$$

The attribute and its cut point, which gives maximum Information Gain are chosen for the testing in the node. The process is recursively repeated to build the child nodes [1].

2.1 Problems with standard decision trees

Some Limitations of the standard decision trees are described below.

To induce a decision tree for a continuous valued data, using the above discussed method, each attribute needs $n - 1$ evaluations (where n is the number of instances, assuming that an attribute is having n distinct values.), to determine the optimal cut point at which information gain is maximum. Therefore it is relatively expensive to work with large data sets. Here, we followed Fayyad [10] idea, which has shown only the class boundary points are to be examined to

Table 1. A toy dataset.

Height (cm)	Weight (kg)	Class-label
52	125	tiger
62	155	tiger
73	170	tiger
18.4	17.9	cat
42	108.2	tiger
25.2	30.5	tiger
18	15	cat
22	24	cat
19.2	23.5	cat
24.9	28.3	tiger
24.7	28.2	tiger

determine the optimal cut point which gives the maximum information gain for the attribute. A cut point is called a boundary cut point, if it falls between a pair of successive instances that belong to two different classes. *Here onwards cut point means that which lies at a class boundary.* Even though, in practice, this significantly reduces the number of evaluations, its worst case time complexity still is $O(n)$.

Another limitation is discussed with an illustrative example. Table 1 shows a toy data belonging to a two class problem which has two attributes *height* and *weight* and two class labels *tiger* and *cat*. Let the crisp rule induced (by applying the method described in Section 2) to the toy data is as follows,

IF *height* > 23.35 THEN *class-label* = *tiger*.

This rule works well in classification for the instances that have no noise.³ Noisy instances are more likely to present in the training data while reading feature values such as height, which might decrease performance of the hard decision tree. Hence, crisp boundary based (cut point) decision rules do not resolve uncertainties in the data. In order to overcome this, soft decision tree based on the fuzzy set theory is used and is explained in the next section.

3. Soft decision tree

This section explains basic concepts of fuzzy set theory followed by its application to standard decision tree called design of soft decision tree.

3.1 Fuzzy set theory

The definitions of crisp set, fuzzy set, crisp partition and fuzzy partition are defined below. Details can be found in [47]. Let O be the collection of objects in the Universe of discourse.

²This paper limits its scope to continuous valued attributes only.

³One example is, the height measuring device has erroneously recorded the height value to be 23.4, when in fact it is 23.34.

Crisp Set: A Crisp set C is a subset of the Universe of discourse O and it is expressed with a sharp characteristic (membership) function $\mu_C : O \rightarrow \{0, 1\}$. For each object $o \in O$,

$$\mu_C(o) = \begin{cases} 1 & \text{if } o \in C, \\ 0 & \text{otherwise.} \end{cases}$$

Fuzzy Set: A Fuzzy set F is a subset of the Universe of discourse O , whose characteristic (membership) function is $\mu_F : O \rightarrow [0, 1]$. For each object $o \in O$,

$$\mu_F(o) = \begin{cases} 1 & \text{if } o \in F, \\ x & \text{if } 0 < x < 1, o \text{ is in } F \text{ with a degree or membership value } x, \\ 0 & \text{otherwise, i.e., } o \notin F. \end{cases}$$

Partition: A hard or crisp partition of the set O is the non-empty set of blocks $\{B_1, B_2, \dots, B_p\}$, where each block is a non-empty crisp set,

$$\begin{aligned} B_i &\subseteq O \text{ for } 1 \leq i \leq p, \\ B_i \cap B_j &= \phi \text{ and} \\ \bigcup_{i=1}^p B_i &= O. \end{aligned}$$

In other words, it is a mutually disjoint and collectively exhaustive collection of blocks.

Fuzzy Partition: A fuzzy or soft partition of O is the non-empty set of fuzzy sets $\{F_1, F_2, \dots, F_p\}$, where each fuzzy set, also known as fuzzy block, is non-empty (i.e., there is an element whose membership value is other than 0), and for each $o \in O$,

$$\sum_{i=1}^p \mu_{F_i}(o) = 1.$$

Fuzzy blocks of a fuzzy partition can be given human understandable names called linguistic variables. For example the height may be divided into two fuzzy blocks called “short” and “tall”. Then a rule can be stated as

IF height is “tall” THEN class-label = tiger.

The degree by which the height is in the fuzzy block “tall” determines the strength of this rule.

This rule is better understandable, hence has good interpret-ability than the rule:

IF height > 23.35 THEN class-label = tiger.

Figure 1 illustrates the above mentioned example. In case of the soft partition, suppose that the height given is 23.4 there is a scope for the final class-label to be either cat or tiger. Whereas in the hard partition, one of this is ruled out.

3.2 Design of soft decision tree

The building process of a soft decision tree is similar to that of the standard decision tree (as discussed in Section 2) except, the way in which the goodness measure is calculated that uses the fuzzy membership values.

Let B be the continuous valued attribute and let τ be the optimal threshold or cut point chosen among all possible thresholds (as discussed in Section 2.1). In case of crisp partition, if attribute B value is less than or equal to a cut point or threshold τ then it belongs to block B_1 else it belongs to block B_2 as shown in figure 1(a). In case of soft partition, if attribute B value is less than or equals to $\tau - w/2$ then it belongs to fuzzy block B_1 called with a linguistic variable “short” else if its value is greater than or equal to $\tau + w/2$ then it belongs to fuzzy block B_2 called with a linguistic variable “tall”, otherwise it belongs to both fuzzy blocks “short” and “tall” with some fuzzy membership values as shown in figure 1(b) where w is overlapping width of two fuzzy blocks “short” and “tall”.

The fuzzy membership value can be calculated as follows. Let $\mu_{B_j}(x_i)$ represents the fuzzy membership of instance x_i to fuzzy block B_j . If an instance x_i belongs to fuzzy block “short” (shown in figure 1(b)) then the fuzzy membership value $\mu_{short}(x_i)$ is calculated as given below.

$$\mu_{short}(x_i) = \begin{cases} 1 & \text{if } x_i \leq \tau - w/2, \\ \frac{(\tau + \frac{w}{2}) - x_i}{w} & \text{if } \tau - \frac{w}{2} < x_i < \tau + \frac{w}{2}, \\ 0 & \text{if } x_i \geq \tau + w/2. \end{cases}$$

Similarly, the membership value $\mu_{tall}(x_i)$ of an instance x_i belongs to fuzzy block “tall” (shown in figure 1(b)) is calculated as given below.

$$\mu_{tall}(x_i) = \begin{cases} 0 & \text{if } x_i \leq \tau - w/2, \\ \frac{x_i - (\tau - \frac{w}{2})}{w} & \text{if } \tau - \frac{w}{2} < x_i < \tau + \frac{w}{2}, \\ 1 & \text{if } x_i \geq \tau + w/2. \end{cases}$$

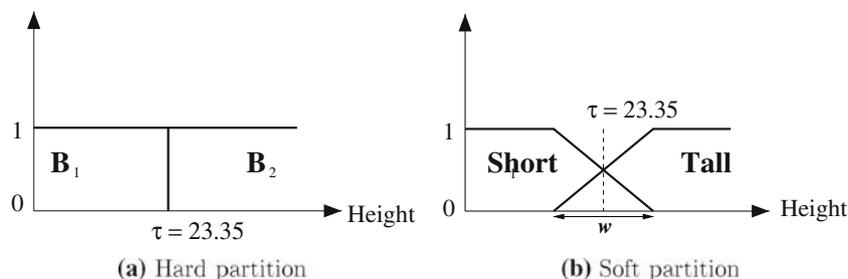


Figure 1. Hard partition Vs soft partition.

In the process of designing a soft decision tree, the fuzzy entropy of a training set \mathcal{X} is calculated as follows:

$$Entropy_F(\mathcal{X}) = \sum_{k=1}^c -(\mu_{p_k}, \mathcal{X}) \log_2(\mu_{p_k}, \mathcal{X}), \quad (4)$$

where $(\mu_{p_k}, \mathcal{X}) = \sum_{x_i \in c_k} (\mu_{B_1}(x_i) + \mu_{B_2}(x_i))$ is the fuzzy proportion of instances in fuzzy set \mathcal{X} that belongs to class c_k . After that, the weighted average of fuzzy entropy or fuzzy info for fuzzy partition of tuples in \mathcal{X} over the values of an attribute B using the optimal threshold τ_k is given below.

$$Info_F(B, \tau_k, \mathcal{X}) = \frac{N_F^{\mathcal{X}_1}}{N_F^{\mathcal{X}}} Entropy_F(\mathcal{X}_1) + \frac{N_F^{\mathcal{X}_2}}{N_F^{\mathcal{X}}} Entropy_F(\mathcal{X}_2), \quad (5)$$

where $Entropy_F(\mathcal{X}_1)$ and $Entropy_F(\mathcal{X}_2)$ are the fuzzy entropy of subsets \mathcal{X}_1 and \mathcal{X}_2 respectively, $N_F^{\mathcal{X}} = \sum_{i=1}^{|\mathcal{X}|} (\mu_{B_1}(b_i) + \mu_{B_2}(b_i))$, $N_F^{\mathcal{X}_1} = \sum_{i=1}^{|\mathcal{X}_1|} \mu_{B_1}(b_i)$, $N_F^{\mathcal{X}_2} = \sum_{i=1}^{|\mathcal{X}_2|} \mu_{B_2}(b_i)$. Similar to the standard decision tree, the fuzzy information gain of an attribute B using a threshold τ_k can be calculated as given below.

$$G_F(B, \tau_k) = Entropy_F(\mathcal{X}) - Info_F(B, \tau_k, \mathcal{X}). \quad (6)$$

Similarly we find fuzzy information gain for all attributes in \mathcal{X} , then we choose the attribute which has high information gain as the good splitting attribute at the root node. The above process is recursively repeated on the obtained child nodes from the root node, to enhance the soft decision tree.

Figure 2 shows a soft decision tree model, it has two test nodes with attributes A and B and three terminal nodes. The inference of a soft decision tree can be discussed as follows. To classify an unseen instance in this soft decision tree model the matching fuzzy membership values of the instance to each node from root to leaf are calculated. Let x_i be the instance to be classified to one of the class labels C_1 and C_2 , then the fuzzy membership value of x_i belonging to each class can be calculated as given below.

- The fuzzy membership value π_1 of the instance x_i to classify to class C_1 is $\mu_{A_1}(x_i) \otimes \mu_{B_1}(x_i)$. (where \otimes is the fuzzy product operation.)
- The fuzzy membership value π_2 of the instance x_i to classify to class C_2 is either $\mu_{A_1}(x_i) \otimes \mu_{B_2}(x_i)$ or $\mu_{A_2}(x_i)$.

The fuzzy product operation of $\mu_{A_1}(x_i)$ and $\mu_{B_1}(x_i)$ denoted $\mu_{A_1}(x_i) \otimes \mu_{B_1}(x_i)$, is $\min(\mu_{A_1}(x_i), \mu_{B_1}(x_i))$.

If the instance x_i is belonging to two classes C_1 and C_2 with fuzzy membership values π_1 and π_2 respectively then class C_1 is assigned to x_i if $\pi_1 > \pi_2$, otherwise class C_2 is assigned.

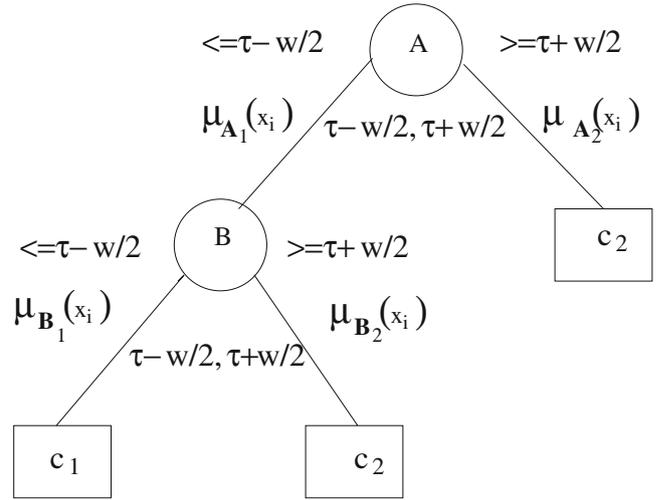


Figure 2. Soft decision tree.

4. A randomized soft decision tree

In this section, the proposed method, a randomized soft decision tree classification model is described and also an ensemble of randomized soft decision trees is presented.

In this method, several cut points are considered for each attribute in the induction process of a decision tree, whereas in the existing method a single cut point is used [21]. In the induction process, at a node, the attribute and its cut point are randomly chosen from distributions as described below.

Let $G_F^{max}(A_i)$ be the maximum information gain for the attribute A_i among its various cut points. The probability distribution over the set of attributes is defined as follows, from which randomly an attribute is chosen.

$$P(A_i) = \frac{G_F^{max}(A_i)}{\sum_{k=1}^d G_F^{max}(A_k)}. \quad (7)$$

Among various cut points of the attribute A_i , the cut point that is chosen is randomly selected from the distribution where the probability of choosing a cut point that corresponds to threshold τ_k is

$$P(\tau_k) = \frac{G_F(A_i, \tau_k)}{\sum_{\forall l} G_F(A_i, \tau_l)}. \quad (8)$$

To avoid over-fitting, pruning is done, which could be either *prepruning* or *postpruning* [1]. The paper uses *prepruning*, where the building process is terminated as soon as the resulting error falls below a prespecified threshold. The error threshold used is chosen by using a three-fold cross validation from $\{0.1, 0.2, 0.3, 0.4, 0.5\}$.

4.1 An ensemble of randomized soft decision trees

There are two frameworks to build an ensemble of classifiers which are, based on a dependent framework or an

Table 2. Experimental results before injecting noise.

	1999 KDDCUP	SPAM MAIL	PIMA
Standard decision tree	94.67% \pm 2.59	93.72% \pm 0.53	72.44% \pm 2.51
Soft decision tree	95.94% \pm 2.1	94.98% \pm 0.46	73.66% \pm 2.1
Randomized soft decision tree	96.98% \pm 1.21	96.26% \pm 0.27	74.89% \pm 1.62
Ensemble of randomized soft DTs	98.23% \pm 0.91	98.14% \pm 0.26	76.96% \pm 1.21

independent framework. In the dependent one, the classifiers are dependent so that the output of one classifier is used in the design process of the next one (*the good example is Adaboost*). Alternatively, parameters used to build the classifier are different from the other components, so that each classifier is independent from the other one (*the good example is Bagging*) [48].

Let \mathcal{X} be the given training set having n tuples. The bootstrap method is applied to derive the sub-training set \mathcal{X}_i (drawn using sampling with replacement). Let T_1, T_2, \dots, T_l be the learned soft decision trees in the ensemble method as discussed in Section 4. Each node is built based on a bootstrapped training set. The parameter l is chosen from a three-fold cross validation from $\{1, 3, 5, 7, 9, 11, 13, 15\}$. For the given query pattern Q , the proposed ensemble model takes output of each randomized soft decision tree T_i , for $i = 1, 2, \dots, l$ and outputs the class label based on majority voting (either 1 or 0) which is assigned to the query pattern Q .

5. Data sets

Three data sets are used in the experimental study.

1. 1999 KDD Cup data set, which is originated by MIT Lincoln labs [49]. Since 1999, KDD Cup data set has been widely for the evaluation of intrusion detection systems. And it was prepared based on the data captured from 1998 DARPA IDS [50] evaluation program. The 1999 KDD Cup training data set was derived from around 4,900,000 connections, each connection is represented as a vector having 41 features, and class information is labeled as either normal or an anomalous one. Among 41 features we used continuous attributes only for our experiments. And the detailed explanation of each continuous feature is given in [51].
2. Spam mail data set collected at Hewlett-Packard Labs. Totally it has 4601 instances with 57 continuous attributes and a nominal class label which categorizes the email as

either a spam one or not. Its documentation and data sets are available at the UCI Machine Learning Repository [52].

3. Pima Indians Diabetes Database originated by National Institute of Diabetes, Digestive and Kidney Diseases. This data set is having 768 instances, each with eight attributes. All eight attributes are continuous valued attributes and details can be found at UCI Machine Learning Repository [52].

6. Experimental results and discussion

In this section, we discussed the results of the proposed model called *ensemble of randomized soft decision trees for robust classification*. The performance of the proposed model is compared against various existing models in terms of accuracy and standard deviation. we used C4.5 package for standard decision tree invented by Quinlan [53]. To evaluate various methods used in this paper, we injected noise at various levels ranging from 1% to 6% in three specified data sets. Table 2 and table 3 have shown the comparison of experimental results over the specified data sets before injecting noise and after injecting noise respectively. Figure 3, figure 4 and figure 5 shows the results of PIMA, SPAMMAIL and 1999KDDCUP data sets respectively. It is clear from the results when the percentage of noise is increased the performance of standard decision tree is decreased abruptly for PIMA and SPAMMAIL data sets, whereas the performance is increased up to some extent and decreased later for 1999KDDCUP data set. For the existing soft decision tree model, proposed randomized soft decision tree model and an ensemble of randomized soft decision tree model, the performance is increased first and later decreased slowly.

Experimental results show that our proposed randomized soft decision tree model and an ensemble of randomized soft decision trees perform better for the standard data sets and also more robust to noise than the remaining methods.

Table 3. Experimental results after injecting noise.

	1999 KDDCUP	SPAM MAIL	PIMA
Standard Decision Tree	91.61% \pm 2.77	91.25% \pm 1.71	71.125% \pm 2.27
Soft Decision Tree	93.55% \pm 2.03	92.54% \pm 1.36	72.84% \pm 1.87
Randomized Soft Decision Tree	94.87% \pm 1.56	93.92% \pm 1.23	74.97% \pm 1.18
Ensemble of Randomized Soft DTs	97.68% \pm 0.89	97.56% \pm 0.73	76.26% \pm 0.67

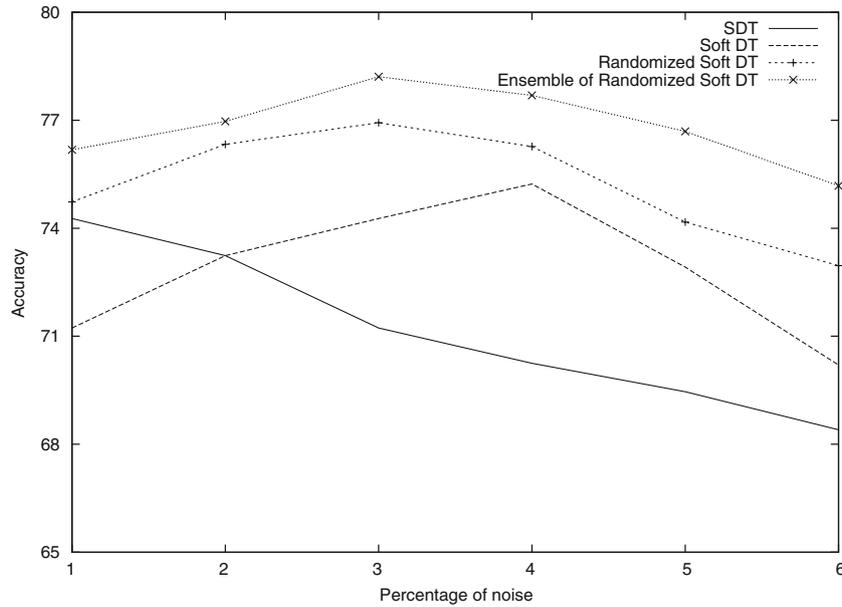


Figure 3. Experimental results over PIMA dataset.

6.1 Complexity of randomized soft decision tree

In this section, the complexity of the proposed method, “A Randomized Soft Decision Tree” is discussed particularly with a single test node, having d number of attributes and n number of tuples. The complexity of an exhaustive search to find optimal cut point requires $n - 1$ evaluations for each attribute and it could be expensive especially as n increases.

The heuristic search of the proposed model needs to examine only class boundary cut points instead of $n - 1$ cut points of each attribute. For k -class problem, when all instances are arranged in the sorted sequence, where all

instances of the same class are adjacent to each other, in the best case $k - 1$ evaluations are used to find the optimal cut point of an attribute. In the worst case, where the class changes from one instance to another, $n - 1$ evaluations are needed for each attribute.

6.2 CPU times

Table 4 gives the idea of computational CPU times of various methods discussed in this paper. These times are recorded on Fedora platform, Intel Core i3 processor with

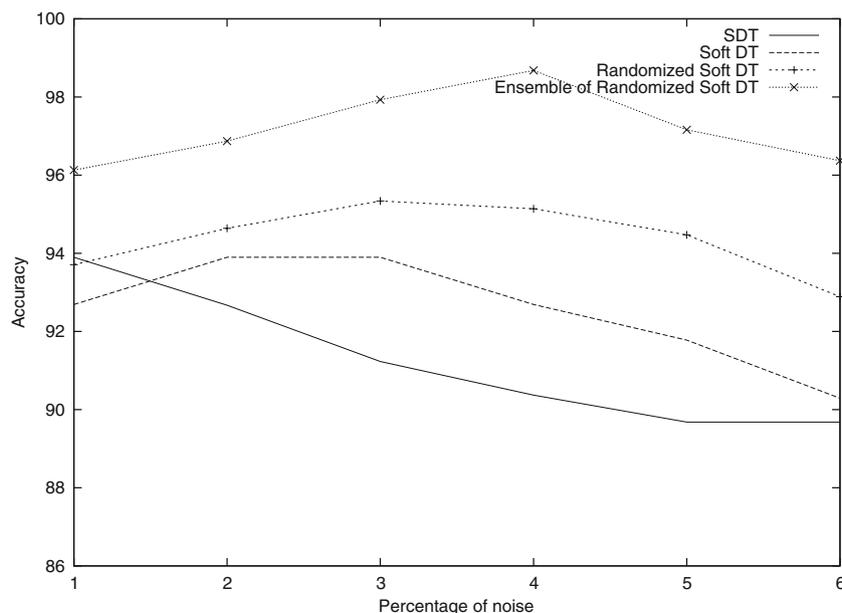


Figure 4. Experimental results over SPAM MAIL dataset.

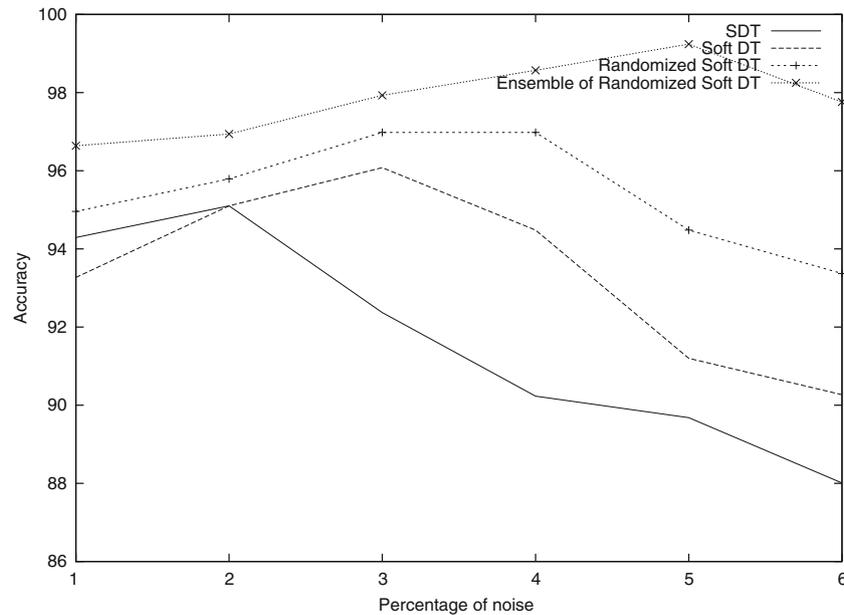


Figure 5. Experimental results over 1999KDDCUP dataset.

Table 4. Comparing CPU times.

Dataset	TRSize	CPU time			
		SDT	Soft DT	Randomized Soft DT	Ensemble of Randomized Soft DT
PIMA	100	0.004s	0.002s	0.002s	0.006s
	200	0.004s	0.002s	0.002s	0.006s
	300	0.008s	0.004s	0.004s	0.011s
	400	0.012s	0.007s	0.007s	0.013s
SPAMMAIL	1000	0.092s	0.068s	0.068s	0.386s
	1500	0.120s	0.082s	0.086s	0.426s
	2000	0.152s	0.098s	0.102s	0.487s
	2500	0.182s	0.136s	0.142s	0.513s
	3000	0.228s	0.158s	0.167s	0.697s
KDDCUP	20000	2.424s	1.812s	1.853s	2.861s
	25000	3.524s	1.941s	1.956s	4.572s
	30000	4.648s	2.256s	2.567s	6.262s
	40000	7.344s	4.124s	4.269s	8.415s

2.40 GHz, 4GB RAM. The observations from table 4 are, the CPU time is increased for standard decision tree (SDT) in all three datasets used in the experiments as the size of training set (TRSize) increases, whereas for the proposed randomized soft decision tree and existing soft decision tree models CPU time is more or less equal as the training set size is increased. For the proposed ensemble method, where each component is derived using the proposed approach i.e., randomized soft decision tree model and its CPU time is increased as the size of the training set increases.

Hence the computational costs involved in the proposed soft decision tree model are better than the existing standard decision tree and it is more or less equal to the existing soft decision tree model.

7. Conclusion

In this paper, the fuzzy set theory is combined with standard decision tree classification to build a randomized soft decision tree model and also an ensemble of randomized soft decision trees for robust classification is presented. For an improvement, instead of information gain as the goodness measure, the parameters like splitting attribute, cut point are randomly chosen from the probability distribution of fuzzy information gain. Experimental results over three standard data sets have shown that the proposed ensemble method and a randomized soft decision tree has outperformed and also more robust classification than the related soft decision tree and also the standard decision tree especially in the presence of noise.

Acknowledgements

Thanks to Suresh Veluru, Department of Engg and Mathematical Sciences, City University London, London, for valuable suggestions.

References

- [1] Han Jiawei and Micheline Kamber 2001 *Data mining: Concepts and techniques*. Academic Press
- [2] Tan Pang-Ning *et al* 2006 *Introduction to data mining*. Pearson Addison Wesley Boston
- [3] Duda Richard O and Peter E Hart 1973 *Pattern classification and scene analysis*. A Wiley-interscience Publication
- [4] Vapnik Vladimir 1999 An overview of statistical learning theory. In: *IEEE Trans. Neural Netw.* 10:988–999
- [5] Zhu Ling 2013 Support vector machine. In: *PSTAT*, pp. 132–135
- [6] Xu Yong *et al* 2013 Coarse to fine K nearest neighbor classifier. In: *Pattern Recognit. Lett.* 34:980–986
- [7] Kolodner Janet 2014 *Case-based reasoning*. Morgan Kaufmann
- [8] Kotsiantis S B 2013 Decision trees: a recent overview. In: *Artif. Intell. Rev.* 39:261–283
- [9] Rodrigo C Barros, Marcio P Basgalupp, Andr I C P L F de Carvalho and Alex A Freitas 2012 A survey of evolutionary algorithms for decision tree induction. In: *IEEE Trans. Syst. Man Cybern.* 42:291–312
- [10] Fayyad and Keki 1992 On the handling of continuous valued attributes in decision tree generation. In: *Mach. Learn.* 8:87–102
- [11] Fayyad and Keki 1993 Multi interval discretization of continuous valued attributes for classification learning. In: *Int. Joint Conf. Artif. Intell.* 93:1022–1027
- [12] Carter and Catlett 1987 Assessing credit card applications using machine learning. In: *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 71–79
- [13] Dietterich and Kong 1995 Machine learning bias, statistical bias and statistical variance of decision tree algorithms. In: Technical Report, Department of Computer Science. Oregon State University, pp. 71–79
- [14] Marsala Christophe 2009 Data mining with ensembles of fuzzy decision trees. In: *IEEE Symposium on Computational Intelligence and Data Mining*, pp. 348–354.
- [15] Cristina Olaru, Louis Wehenkel 2003 A complete fuzzy decision tree technique. In: *Fuzzy Sets Syst.* 138:221–254
- [16] Quinlan 1996 Improved use of continuous attributes in C4.5. In: *Artif. Intell. Res.* 4:77–90
- [17] Buntine 1992 Learning classification trees. In: *Stat. Comput.* 2: 63–73
- [18] Ouzden and William 1993 Induction of rules subject to a quality constraint probabilistic inductive learning. In: *IEEE Transactions on Knowledge and Data Engineering*, pp. 979–984
- [19] Xiaomeng Wang and Christian Borgelt 2004 Information measures in fuzzy decision trees. In: *Fuzzy Syst.* IEEE, pp. 979–984
- [20] Wang Liu, Hong and Tseng 1999 A fuzzy inductive learning strategy for modular rules. In: *Fuzzy Sets Syst.* pp. 91–105
- [21] Peng Yonghong and Peter A Flach 2001 Soft discretization to enhance the continuous decision tree induction. In: *ECML/PKDD Workshop: IDDM*
- [22] Chen Min and Simone A Ludwig 2013 Fuzzy decision tree using soft discretization and a genetic algorithm based feature selection method. In: *2013 World Congress on Nature and Biologically Inspired Computing (NaBIC)*. IEEE, pp. 238–244
- [23] Umamo Okamoto, Hatono and Tamura 1994 Fuzzy decision trees by fuzzy ID3 algorithm and its application to diagnosis systems. In: IEEE, pp. 2113–2118
- [24] Detyniecki and Marsala 2007 Forest of fuzzy decision trees and their application in video mining. In: *Proceedings of the 5th EUSFLAT Conference*. pp. 345–352
- [25] Pradhan Biswajeet 2013 A comparative study on the predictive ability of the decision tree, support vector machine and neuro-fuzzy models in landslide susceptibility mapping using GIS. In: *Comput. Geosci.* 51: 350–365
- [26] Freund and Schapire 1996 Experiments with a new boosting algorithm. In: *Proceedings 13th International Conference on Machine Learning*. (San Francisco). Morgan Kaufmann, pp. 148–146
- [27] Schapire 1990 The strength of weak learnability. In: *Mach. Learn.* 5:197–227
- [28] Zhang Cha and Yunqian Ma 2012 *Ensemble machine learning*. Springer
- [29] Breiman L 1996 Bagging predictors. In: *Mach. Learn.* 24:123–140
- [30] Efron and Tibshirani 1993 An introduction to the bootstrap. In: *Chapman and Hall, CRC Press*
- [31] Freund Iyer, Schapire and Singer 2003 An efficient boosting algorithm for combining preferences. In: *Mach. Learn.* 4:933–969
- [32] Breiman 2001 Random forests. In: *Mach. Learn.* 45: 5–32
- [33] Cunningham Pdraig 2007 Ensemble techniques. In: Techreport
- [34] Geurts Ernst and Wehenkel 2006 Extremely randomized trees. In: *Mach. Learn.* 63:3–42
- [35] Hamza and Larocque 2005 An empirical comparison of ensemble methods on classification trees. In: *Stat. Comput. Simul.* 75:629–643
- [36] Wei Fan, TJ Watson Res and Sheng Ma 2003 Is random model better? On its accuracy and efficiency. In: *Data Mining, 2003. ICDM 2003. Third IEEE International Conference*. IEEE, pp. 51–58
- [37] Wei Fan, TJ Watson Res, Hawthorne and McCloskey J 2005 Effective estimation of posterior probabilities: Explaining the accuracy of randomized decision tree approaches. In: *Data Mining, Fifth IEEE International Conference*. IEEE pp. 51–58
- [38] Amit Yali and Donald Geman 1997 Shape quantization and recognition with randomized trees. In: *Neural computation*. MIT Press, pp. 1545–1588
- [39] Dietterich T 2001 An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, and randomization. In: *Mach. Learn.* 40:5–32
- [40] Marsala and Bouchon-Meunier 1997 Forest of fuzzy decision trees. In: *Seventh International Fuzzy Systems Association World Congress*, pp. 369–374
- [41] Janikow and Faifer 2000 Fuzzy decision forest. In: *19th International Conference of the North American Fuzzy Information Processing Society*, pp. 218–221
- [42] Crockett Bandar and Mclean 2001 Growing a fuzzy decision forest. In: *10th International Conference on Fuzzy Systems*. IEEE, pp. 614–617

- [43] Janikow 2003 Fuzzy decision forest. In: *22nd International Conference of the North American Fuzzy Information Processing Society*, pp. 480–483
- [44] Bonissone Cadenas, Garrido and Diaz-Valladares 2008 A fuzzy random forest: Fundamental for design and construction. In: *12th International Conference on Information Processing and Management of Uncertainty in Knowledge Based Systems*. Malaga Spain, pp. 1231–1238
- [45] Kuncheva 2003 Fuzzy vs non-fuzzy in combining classifiers designed by boosting. In: *IEEE Trans. Fuzzy Syst.* 11:729–741
- [46] Fumera Giorgio and Fabio Roli 2005 A theoretical and experimental analysis of linear combiners for multiple classifier systems. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27
- [47] Zadeh 1965 Fuzzy sets. In: *Inform. Control* 8:338–353
- [48] Lior Rokach 2010 Ensemble-based classifiers. In: *Artif. Intell. Rev.* 33:1–39
- [49] Lippman Fried Graf and Zissman 2000 Evaluating intrusion detection systems: The 1998 DARPA off-line intrusion detection evaluation. In: *Proceedings of DARPA Information Survivability Conf. and Exposition (DISCEX'00)*, pp. 12–26
- [50] DARPA Intrusion Detection Data Sets. URL: <http://www.ll.mit.edu/mission/communications/ist/corpora/ideval/data/index.html>
- [51] Mahbod Tavallaei Ebrahim Bagheri, Wei Lu and Ali Ghorbani 2009 A Detailed analysis of the KDD CUP 99 data set. In: *Proceedings of 2009 IEEE Symposium on Computer Intelligence in Security and Defense Applications (CISDA)*
- [52] Machine Learning Repository. URL: <http://archive.ics.uci.edu/ml>
- [53] Quinlan 1993 *C4.5 programs for machine learning*. Morgan Kaufmann, Los Atlos