An Empirical Study of Fuzzy Decision Tree for Gradient Boosting Ensemble

Zhaoqing Liu[0000−0003−3486−2808], Anjin Liu[0000−0002−0733−7138], Guangquan Zhang[0000−0003−3960−0583], and Jie Lu[0000−0003−0690−4732]

Australian Artificial Intelligence Institute, University of Technology Sydney, Sydney, Australia
Email: Zhaoqing.Liu-1@student.uts.edu.au, {Anjin.Liu, Guangquan.Zhang, Jie.Lu}@uts.edu.au

Abstract. Gradient boosting has been proved to be an effective ensemble learning paradigm to combine multiple weak learners into a strong one. However, its improved performance is still limited by decision errors caused by uncertainty. Fuzzy decision trees are designed to solve the uncertainty problems caused by the collected information's limitation and incompleteness. This paper investigates whether the robustness of gradient boosting can be improved by using fuzzy decision trees even when the decision conditions and objectives are fuzzy. We first propose and implement a fuzzy decision tree (FDT) by referring to two widely cited fuzzy decision trees. Then we propose and implement a fuzzy gradient boosting decision tree (FGBDT), which integrates a set of FDTs as weak learners. Both the algorithms can be set as non-fuzzy algorithms by parameters. To study whether fuzzification can improve the proposed algorithms in classification tasks, we pair the algorithms with their non-fuzzy algorithms and run comparison experiments on UCI Repository datasets in the same settings. The experiments show that the fuzzy algorithms perform better than their non-fuzzy algorithms in many classical classification tasks. The code is available at `github.com/ZhaoqingLiu/FuzzyTrees`.

Keywords: Decision tree · Fuzzy system · Gradient boosting · Classification.

1 Introduction

Fuzzy concepts and fuzzy objectives are ubiquitous in the process of human cognition and decision-making in real life [19]. Therefore, it is one of the hot areas of research to use fuzzy computing methods to deal with information that is not easy to be quantified. As a typical fuzzy algorithm, fuzzy decision trees [29, 33] have been used in many applications, e.g., diagnosis systems [29], healthcare [1] and accounting and audit [2]. A fuzzy decision tree is an extension and improvement of traditional decision trees. A fuzzy decision tree fuzzifies training sets to fuzzy sets in the data preprocessing stage and computes splitting criteria for feature selection in the tree construction stage [33].

Gradient boosting [13,22] is a primary boosting algorithm in ensemble learning. It can effectively improve a single base learning algorithm's performance [13]. The idea of gradient boosting is derived from the gradient descent method and has been widely used [20,21]. Its basic principle is to train the next weak learner according to the negative gradient of the current learner's loss function and then integrate the trained weak learner into a single strong learner in the form of an iterative combination.

Although some studies focus on fuzzy decision trees, no one compares fuzzy gradient boosting decision trees with non-fuzzy gradient boosting decision trees, and few implementations of fuzzy decision trees are based on the CART [3], which is one of the classical decision trees. In this paper, we propose a fuzzy decision tree (FDT) and fuzzy GBDT (FGBDT), which can combine the inductive capability of decision trees with the capability of fuzzy sets [34] to express uncertainty.

Unlike the fuzzy decision trees in other literature, we use the Fuzzy C-Means clustering algorithm (FCM) [9] as a feature fuzzification method in data preprocessing to improve the effectiveness of describing fuzzy concepts. We also use fuzzy Gini impurity-based metric calculation to extend the heuristic search techniques used in traditional decision trees. Compared with the fuzzy entropy [6,34] used in most existing fuzzy decision trees, the fuzzy Gini impurity used in our fuzzy feature selection technique can reduce the computation of splitting criteria and simplify the tree. For an empirical study, we develop the proposed algorithms into a software toolkit in Python. Finally, we carry out experiments, and the results show that the fuzzy trees have more advantages in solving the classification uncertainty than the non-fuzzy trees.

The contributions of this paper are summarised as follows:

- A novel fuzzy decision tree-based gradient boosting algorithm is proposed.
- A Python toolkit for FDT and FGBDT is developed.
- Extensive experiments indicate that FDT and FGBDT can achieve better accuracy in many classification tasks than non-fuzzy FDT and non-fuzzy FGBDT, respectively.

The paper is organised as follows. Section 2 introduces the related work. The details of FDT and FGBDT are presented in Section 3. Section 4 analyses the experimental evaluation. Conclusion and future work are discussed in Section 5.

2 Related Work

In this section, we review the researches on fuzzy decision trees and gradient boosting.

2.1 Fuzzy Decision Trees

As one of the most representative algorithms in machine learning, a decision tree uses a tree-like model of symbols, rules, and logic to represent knowledge and make logical inferences. A fuzzy decision tree is an extension of the classical

decision tree. It is more robust in tolerating uncertainty by introducing fuzzy sets. Most of the fuzzy decision trees proposed in the literature can be regarded as the variants of the ID3 invented by Quinlan [25] and the C4.5 developed by Quinlan [26]. However, few studies involved fuzzy extensions of the CART algorithm introduced by Breiman et al. [3]. Yuan and Shaw [33] proposed a fuzzy decision tree induction method similar to the ID3 [25], except they use the classification ambiguity metric instead of entropy as the heuristic induction criterion. In the method, feature fuzzification in the data preprocessing stage is essential for constructing a fuzzy decision tree. A similar idea was also proposed in Kosko's study [16]. The author uses a simple algorithm to generate a triangular membership function to fuzzify the training sets into fuzzy membership degrees. The Fuzzy ID3 algorithm (Fuzzy ID3) proposed by Umanol et al. [29] uses fuzzy entropy first introduced by Zadeh [34], and the axiom construction of entropy of fuzzy sets was further introduced by De Luca and Termini [6]. Unlike the traditional ID3 algorithm, Fuzzy ID3 used the probability of membership degrees instead of the probability of crisp samples to calculate the metric, i.e., fuzzy entropy, when selecting the optimal splitting feature. In the literature, many other studies have extended entropy based on fuzziness in various ways.

2.2 Gradient Boosting

Gradient boosting is an ensemble learning technique and one of the most popular algorithms. Its basic idea is derived from the gradient descent method [32]. According to Schapire's proof conclusion based on Hoeffding Inequality [14] and the probably approximately correct (PAC) learning model [27], in the case that the errors of weak learners are independent of each other, the error rate of an ensemble method decreases exponentially with the increase of the number of weak learners and eventually tends to zero. Based on the proof, the gradient descent method can combine multiple weak learners in a strong learner. The regression gradient boosting algorithms developed by Freund et al. [12, 13] can be used for regression and classification tasks. A more general view of functional gradient boosting was proposed by Llew et al. [22]. The basic principle of the algorithms is to train the next weak learner according to the negative gradient information of the loss function of the current model in multiple iterations, and then combine the trained weak learner into the model combination in the form of addition and finally minimise the loss function in the function space [13]. Boosting algorithms have made considerable progress in many areas of machine learning and statistics beyond regression and classification. In many current studies, the standard weak learners of gradient boosting algorithms are traditional decision trees, logistic regression classifiers, Naive Bayes, and other non-fuzzy algorithms [24].

3 Fuzzy Decision Trees for Gradient Boosting Ensemble

In this section, we detail two algorithms: FDT and FGBDT.

3.1 Framework of Fuzzy Decision Trees

The tree construction and prediction stages of FDT and its non-fuzzy FDT are similar. At the same time, there are two differences between the two algorithms. One is that the former executes additional feature fuzzification in data preprocessing, and the other is that the former uses the fuzzy sets to calculate all splitting criteria in tree construction. Specifically, the FDT calculates fuzzy metrics according to the fuzzy membership degrees obtained from the feature fuzzification rather than crisp samples.

Feature Fuzzification in Data Preprocessing Feature Fuzzification (FF) refers to the fuzzy transformation of features in the data preprocessing stage before constructing an FDT tree. The membership function used in FF can be obtained from statistical data or determined based on the fuzzy clustering method of self-organising learning [33]. After the transformation based on the membership function, the calculated fuzzy membership degrees of the features belonging to a group of fuzzy sets are added to the samples. Considering that the Fuzzy C-Means clustering algorithm (FCM) [9] is one of the most widely used methods for feature fuzzification in fuzzy decision tree studies, we use it to cluster a group of fuzzy sets from each feature and then to calculate the fuzzy membership degrees of each feature belonging to these fuzzy sets. By calculating the membership matrix, the FCM makes the features with the maximum similarity be grouped in the same cluster, while the ones with the minimum similarity are divided into different clusters. Also, FF is a prerequisite for constructing FDTs because the metric fuzzification in the tree construction stage is based on the fuzzy membership degrees generated by FF.

Metric Fuzzification in Tree Construction Metric fuzzification (MF) refers to the fuzzy calculation of splitting criteria for feature selection in the tree construction stage. We take the fuzzy metrics, also known as fuzzy entropy and fuzzy information gain, proposed by Umanol et al. [29] into our algorithm framework for feature selection. Specifically, assume that we have a set of samples S, where each sample has ℓ numerical features $A_1, A_2, ..., A_\ell$, and a labelled class $C = \{C_1, C_2, ..., C_n\}$ and fuzzy sets $F_{i1}, F_{i2}, ..., F_{im}$ for the feature A_i (the value of m usually varies from feature to feature). The fuzzy entropy of a fuzzy set of S is defined by:

$$I(S) = -\sum_{k=1}^{n}(p_k \log_2 p_k), \quad \text{where } p_k = \frac{\left|S^{C_k}\right|}{|S|}, \tag{1}$$

here S^{C_k} is a fuzzy subset of S, labelled as the class C_k, $\left|S^{C_k}\right|$ is the sum of the fuzzy membership degrees in S^{C_k}, and $|S|$ is the sum of the fuzzy membership degrees in a fuzzy set of S. Then, the fuzzy information gain $G(A_i, S)$ for A_i by a fuzzy set of S is defined by:

$$G(A_i, S) = I(S) - E(A_i, S), \quad \text{and} \quad E(A_i, S) = \sum_{j=1}^{m}(p_{ij} I(S_{F_{ij}})), \tag{2}$$

where

$$p_{ij} = \frac{\left|S_{F_{ij}}\right|}{\sum_{j=1}^{m}\left|S_{F_{ij}}\right|}, \tag{3}$$

$S_{F_{ij}}$ is a fuzzy subset split from a fuzzy set of S on A_i, and $\left|S_{F_{ij}}\right|$ is the sum of the fuzzy membership degrees in $S_{F_{ij}}$.

Although the fuzzy FD3 still adopts a greedy strategy to construct a tree, the two metrics used for the greedy strategy are different from the classic probability-based entropy and information gain.

- The first metric is fuzzy entropy, which is defined by Eq. 1. In Eq. 1, p_k equals the proportion of the sum of the fuzzy membership degrees in a fuzzy subset S^{C_k} of S to the sum of the fuzzy membership degrees in a fuzzy set of S. By comparison, according to Shannon [28], the classic information entropy can be given by $I(S) = -\sum_{k=1}^{n} p_k \log_2 p_k$, where S represents a set of samples with n classes; suppose $k \in \{1, 2, ..., n\}$, p_k is the probability of the samples labelled with class k in S.
- The second metric is the fuzzy information gain, which is defined by Eq. 2 and Eq. 3. In the two equations, p_{ij} equals the proportion of the sum of the fuzzy membership degrees in a fuzzy subset $S_{F_{ij}}$, which is split from a fuzzy set of S on A_i to the sum of the fuzzy membership degrees in the fuzzy set of S. In contrast, according to Quinlan [25], the classic information gain for an feature $a \in A_i$ can be given by $G(S, A_i) = I(S) - I(S|A_i) = I(S) - \sum_{a \in A_i} p(a) \sum_{i=1}^{n} -Pr(i|a) \log_2 Pr(i|a)$, where $I(S)$ represents the entropy of the parent node, $I(S|A_i)$ represents the weighted sum of the entropy of the child nodes split from S on A_i; and $Pr(i|a)$ is the conditional probability of Pr_i given a.

By analogy, we introduce two new concepts: fuzzy Gini impurity and fuzzy information gain ratio into the algorithm framework to support FDT and other fuzzy decision trees. The two concepts are the fuzzy extensions of the Gini impurity used in Breiman et al. [3] and the information gain ratio used in Quinlan [25]. Specifically, the fuzzy Gini impurity of a fuzzy set S is defined by:

$$I_G(S) = \sum_{k=1}^{n} p_k(1 - p_k), \tag{4}$$

where p_k is given by Eq. 1. Then, the fuzzy information gain ratio for the feature A_i by S is defined by:

$$GR(A_i, S) = \frac{G(A_i, S)}{IV(A_i, S)}, \tag{5}$$

where $G(A_i, S)$ is given by Eq. 2, and $IV(A_i, S)$ is the intrinsic value of A_i. That is, suppose A_i has a set of all possible values $V = \{V_1, V_2, ..., V_q\}$,

$$IV(A_i, S) = -\sum_{t=1}^{n}(p_t \log_2 p_t), \quad \text{where} \quad p_t = \frac{\left|A_i^{V_t}\right|}{|A_i|}, \tag{6}$$

here $A_i^{V_t}$ is a fuzzy subset of A_i with value V_t, $\left|A_i^{V_t}\right|$ is is the sum of the fuzzy membership degrees in $A_i^{V_t}$, and $|A_i|$ is the sum of the fuzzy membership degrees in a fuzzy set of A_i.

The pseudocode for FDT is shown in Algorithm 1.

Algorithm 1 FDT algorithm.

1: **if** the current node at the level L with a fuzzy set of S satisfies: the current sample size is less than the threshold $min_{samples_{split}}$, or the tree's current depth is greater than the threshold max_{depth}, or no features available for splitting tests. **then**
2: The current node is a leaf and is assigned a class label by majority vote calculation in classification tasks and a numerical label by mean calculation in regression tasks.
3: **else**
4: **for** each $A_i(i = 1, 2, ...\ell)$ **do**
5: Split S into two subsets.
6: Calculate the information gain according to the Eq. (2), where the Eq. (1) is replaced by (4).
7: Select the test feature $Amax$ that maximises the information gain.
8: **if** the test *best_impurity_gain* is greater than the threshold *min_impurity_split* **then**
9: Generate a node according to the set of samples and the corresponding fuzzy sets containing their fuzzy membership degrees.
10: Make the current level $L = L + 1$.
11: Repeat recursively from Line 1 for both subsets, respectively.
12: **end if**
13: **end for**
14: **end if**

3.2 Implementation of Gradient Boosted Fuzzy Decision Trees

FGBDT combines multiple weak fuzzy learners into a single strong learner in an iterative fashion, then gradually approximate the optimal learner in a greedy fashion [13]. The main difference is that it integrates a set of regression FDTs instead of non-fuzzy regression trees. According to Vapnik's empirical risk minimisation principle [31], the algorithm iteratively performs the optimisation using a function gradient descent method, i.e., the steepest descent step. Specifically, the algorithm still uses the first-order derivative of the loss function to generate a set of pseudo residuals, namely the first-order Taylor polynomial in Taylor's theorem, to determine the loss function's steepest gradient descent in the current function space. Then the algorithm modifies the learner through the negative gradient direction to make it better. Algorithm 2 presents the pseudocode for the generic FGBDT for regression.

Algorithm 2 FGBDT Algorithm.

Input: Training set $\{(x_i, y_i)\}_{i=1}^{N}$; a differentiable loss function $L(y, f(x))$; number of iterations M;

1: Initialise a single learner with a constant value: $f_0(x) = \arg\min_\gamma \sum_{i=1}^{N} L(y_i, \gamma)$.
2: **for** $m = 1$ to M **do**
3: Iteratively calculate the pseudo residuals corresponding to x:
4: **for** $i = 1, 2, ..., N$ **do**
5: $r_m = -\left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f(x)=f_{m-1}(x)}$.
6: **end for**
7: Fit a regression FDT to the pseudo residuals r_{i_m} giving disjoint regions $R_{j_m}, j = 1, 2, ..., J_m$, that is, using the training set $\{(x_i, r_i m)\}_{i=1}^{N}$.
8: Use the fitted regression FDT to calculate the pseudo residuals by the gradient descent method: $\gamma_m = \arg\min_\gamma \sum_{i=1}^{N} L(y_i, f_{m-1}(x_i) + \gamma h_m(x_i))$.
9: Update the learner $f_m(x) = f_{m-1}(x) + \gamma_m h_m(x)$ for the optimisation in the next iteration.
10: **end for**

Output: $\hat{f}(x) = f_M(x)$.

3.3 Time Complexity

The time complexity of the two algorithms is shown in Table 1.

Table 1: Time complexity for FDT and FGBDT.

Algorithm	Time complexity	
	Training	Prediction
FDT	$\mathcal{O}(N \log_2 NMC) \sim \mathcal{O}(N^2 MC)$	$\mathcal{O}(\log_2 N) \sim \mathcal{O}(N)$
FGBDT	$\mathcal{O}(TN \log_2 N) \sim \mathcal{O}(TN^2)$	$\mathcal{O}(T \log_2 N) \sim \mathcal{O}(TN)$

For tree training, the average depth of an FDT tree is $\log_2 N$, where N is the number of samples, and the worst-case depth is N, so the time complexity for the depth is between $O(\log_2 N)$ and $O(N)$. In a tree construction, FDT calculates the Gini impurity or entropy of the samples based on the current sample size and then calculates the information gain or information gain ratio in each iteration. So the time complexity for the tree is between $O(N \log_2 NM)$ and $O(N^2 M)$, where M is the number of features. In calculating each information gain and information gain ratio, FDT calculates the sum of membership degrees of features of the current node and the sum of membership degrees of features in each of its sub-trees. Therefore, the overall time complexity for the tree is between $O(N \log_2 NMC)$ and $O(N^2 MC)$, where C is the number of fuzzy sets (i.e., clusters) of a feature. Also, the time complexity for FGBDT is between $O(TN \log_2 N)$ and $O(TN^2)$, where T is the number of integrated trees. For prediction, the time complexity for FDT is between $O(\log_2 N)$ and $O(N)$, and that for FGBDT is between $O(T \log_2 N)$ and $O(TN)$.

According to the Master theorem, because $M \ll N$, $C \ll N$ and $T \ll N$, the time complexity for the training of both algorithms is $\mathcal{O}(N \log_2 N) \sim \mathcal{O}(N^2)$, and that for the prediction of both algorithms is $\mathcal{O}(\log_2 N) \sim \mathcal{O}(N)$.

4 Experimental Evaluation

This section empirically evaluates the proposed FDT and FGBDT.

4.1 Datasets

We consider six datasets from the UCI Machine Learning Repository [8].
• **Vehicle Silhouettes** (**VS**) [8]. The task is to classify a given silhouette as one of four types of vehicle, using a set of features extracted from the silhouette (846 samples, 18 features, and 4 classes).

• **German Credit** (**GC**) [5]. The task is to classify people described by a set of features as good or bad credit risks (1, 000 samples, 24 features, and 2 classes).

• **Pima Indians Diabetes** (**PID**) [30]. The task is to classify the diabetes tests based on the features of a group of people (768 samples, 8 features, and 2 classes).

• **Iris** [17]. The task is to classify the types of iris plant based on the attributes of the four flowers (150 samples, 4 features, and 3 classes).

• **Wine** [18]. The task is to classify the origin of wines using the results of chemical analysis (178 samples, 13 features, and 3 classes).

• **Forest Cover Type** (**FCT**) [21]. The task is to classify forest cover type from cartographic variables (581, 012 samples, 54 features, and 7 classes).

4.2 Baselines and Experimental Setup

We conduct four comparison experiments. The first two are to study the effect of feature fuzzification (FF) only and the combination of FF and metric fuzzification (MF) on the performance of FDT; the second is to investigate the impact of the above two types of fuzzification on the performance of FGBDT; the last one is to compare the performance of FDT with published baselines. In the first experiment, the non-fuzzy FDT (DT) without FF is the baseline of the non-fuzzy FDT with FF. In the other two experiments, the non-fuzzy FDT and non-fuzzy FGBDT are the baselines of FDT and FGBDT. In the last experiment, we compare our FDT with six representative baselines: XGBClassifier (XGBoost) [4], CatBoostClassifier (CatBoost) [7], LGBMClassifier (LightGBM) [15], HoeffdingTreeClassifier (HT) [23], HoeffdingAdaptiveTreeClassifier (HAT) [23] and SAMKNNClassifier (SAMKNN) [23].

For all datasets, categorical features are transformed into numeric features. For hyperparameters, $disable\_fuzzy$ for each classifier is used to specify whether to use fuzzy rules, and max_depth is set to 5; $learning\_rate$ for each pair of fuzzy and non-fuzzy FGBDT classifiers is set to 0.1, and $n\_estimators$ is set to 100; all other hyperparameters are left as their default values; n_conv for each FCM transformer is set to one of $\{3, 4, 5\}$ and 5 by default to specify the number of fuzzy sets to generate. Also, we quantify the average performance of each classifier through 10-round 10-fold cross-validation training and testing in each experiment. In the last experiment, we randomly select $1,000$ samples from the dataset FCT for the training and testing of FDT and baseline classifiers. The same hyperparameters for all tree classifiers are set to identical values, except that the hyperparameters for the SAMKNN classifier are left as their default values. Our code is available on GitHub [1].

Table 2: Results with FF (n_conv = 3) and without FF.

Task	DT with FF		DT without FF	
	Acc	Std	Acc	Std
VS	**0.6963**	0.0341	0.6643	0.0305
GC	0.7080	0.0449	**0.7100**	0.0316
PID	**0.7226**	0.0478	0.7084	0.0509
Iris	**0.9333**	0.0629	**0.9333**	0.0629
Wine	0.8935	0.0676	**0.8990**	0.0742
Avg	**0.7907**	0.0515	0.7830	0.0500

Table 3: Results with FF (n_conv = 4) and without FF.

Task	DT with FF		DT without FF	
	Acc	Std	Acc	Std
VS	**0.6963**	0.0341	0.6643	0.0305
GC	0.7080	0.0449	**0.7100**	0.0316
PID	**0.7226**	0.0478	0.7084	0.0509
Iris	**0.9333**	0.0629	**0.9333**	0.0629
Wine	0.8935	0.0676	**0.8990**	0.0742
Avg	**0.7907**	0.0515	0.7830	0.0500

Table 4: Results with FF (n_conv = 5) and without FF.

Task	DT with FF		DT without FF	
	Acc	Std	Acc	Std
VS	**0.6963**	0.0341	0.6643	0.0305
GC	0.7080	0.0449	**0.7100**	0.0316
PID	**0.7226**	0.0478	0.7084	0.0509
Iris	**0.9333**	0.0629	**0.9333**	0.0629
Wine	0.8935	0.0676	**0.8990**	0.0742
Avg	**0.7907**	0.0515	0.7830	0.0500

Table 5: Results with FDT and non-fuzzy FDT.

Task	DT with PF		DT with non-PF	
	Acc	Std	Acc	Std
VS	**0.6915**	0.0619	0.6643	0.0305
GC	**0.7200**	0.0287	0.7100	0.0316
PID	**0.7422**	0.0389	0.7084	0.0509
Iris	**0.9333**	0.0629	**0.9333**	0.0629
Wine	**0.9108**	0.0650	0.8990	0.0742
Avg	**0.7996**	0.0515	0.7830	0.0500

[1] https://github.com/ZhaoqingLiu/FuzzyTrees

Table 6: Results with FGBDT and non-fuzzy FGBDT.

Task	DT with PF		DT with non-PF	
	Acc	Std	Acc	Std
VS	**0.6832**	0.0457	0.6572	0.0223
GC	**0.6840**	0.0504	0.6790	0.0451
PID	**0.7082**	0.0465	0.7031	0.0642
Iris	**0.9400**	0.0663	0.9333	0.0629
Wine	0.8987	0.0692	**0.8990**	0.0742
Avg	**0.7828**	0.0556	0.7743	0.0537

Table 7: Comparison between FDT and baselines on dataset FCT.

Methods	Acc	Std
XGBoost_b	0.6566	0.0395
CatBoost_b	0.6302	0.0455
LightGBM_b	0.4880	0.0518
HT	0.5390	0.0524
HAT	0.5418	0.0510
SAMKNN	0.5533	0.0600
FDT	**0.6639**	0.0434

4.3 Experimental Results

Non-fuzzy FDT with FF vs. Non-fuzzy FDT without FF Table 2, 3 and 4 summarise the prediction accuracy (Acc) and standard deviation (Std) of three pairs of non-fuzzy FDT trained on samples with and without FF, respectively. The results show that the average performance (Avg) of FDT with FF is better than that without FF.

FDT vs. Non-fuzzy FDT Table 5 shows the respective Acc and Std for FDT and non-fuzzy FDT. We observe that the FDT with the combination of FF and MF outperform the non-fuzzy FDT on the average performance. In other words, using fuzzy sets to quantify fuzzy objects in classification tasks can help FDT improve performance (nearly 1.7%).

FGBDT vs. Non-fuzzy FGBDT As shown in Table 6, the FGBDT outperforms the non-fuzzy FGBDT. We consider that FGBDT with FF and MF can further enhance the optimisation (nearly 0.9%) of gradient boosting by using the fuzzy membership degrees added to the samples as the fuzzy rules.

FDT vs. State-of-the-Art We also compare FDT with six state-of-the-art algorithms. For a fair comparison, we only take the base learner of the ensemble algorithms for comparison. Table 7 show that FDT yields significant performance improvement compared with the six representative baselines.

5 Conclusion and Future Work

We have proposed and implemented a fuzzy gradient boosting algorithm framework with FDT and FGBDT. We have also conducted three comparison experiments to study how fuzzification affects the algorithms' performance. In conclusion, FGBDT can improve performance and enhance gradient boosting's optimisation effect in many classification tasks.

Based on the current research presented in this paper, we will further use FCM to identify the appropriate membership functions, optimise multiple ensemble learning methods based on FDT, and study how to reduce the computational overhead of the current algorithms. Furthermore, based on our observation, fuzzy sets theory does not consistently outperform the classic method. We consider this problem is caused by over fuzzification. Therefore, how to design a proper fuzzification controlling mechanism is also worth studying. Also, using transfer learning [10, 11, 35] to generate decision trees is an exciting challenge.

Acknowledgment

The work presented in this paper was supported by the Australian Research Council (ARC) under Discovery Project DP190101733.

References

1. Armand, S., Watelain, E., Roux, E., Mercier, M., Lepoutre, F.X.: Linking clinical measurements and kinematic gait patterns of toe-walking using fuzzy decision trees. Gait & Posture **25**(3), 475–484 (2007)
2. Beynon, M.J., Peel, M.J., Tang, Y.C.: The application of fuzzy decision tree analysis in an exposition of the antecedents of audit fees. Omega **32**(3), 231–244 (2004)
3. Breiman, L., Friedman, J., Stone, C.J., Olshen, R.A.: Classification and regression trees. CRC press (1984)
4. Chen, T., Guestrin, C.: Xgboost: A scalable tree boosting system. In: SIGKDD. pp. 785–794. ACM (2016)
5. Chen, Z.: The application of tree-based model to unbalanced german credit data analysis. In: MATEC Web of Conferences. vol. 232, p. 01005. EDP Sciences (2018)
6. De Luca, A., Termini, S.: A definition of a nonprobabilistic entropy in the setting of fuzzy sets theory. Information and Control **20**(4), 301–312 (1972)
7. Dorogush, A.V., Ershov, V., Gulin, A.: Catboost: gradient boosting with categorical features support. CoRR **abs/1810.11363** (2018)
8. Dua, D., Graff, C.: UCI machine learning repository (2017), `http://archive.ics.uci.edu/ml`
9. Dunn, J.C.: A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. Journal of Cybernetics (1973)
10. Fang, Z., Lu, J., Liu, F., Xuan, J., Zhang, G.: Open set domain adaptation: Theoretical bound and algorithm. IEEE Transactions on Neural Networks and Learning Systems (2019)
11. Fang, Z., Lu, J., Liu, F., Zhang, G.: Unsupervised domain adaptation with sphere retracting transformation. In: IJCNN. pp. 1–8. IEEE (2019)
12. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences **55**(1), 119–139 (1997)
13. Friedman, J.H.: Greedy function approximation: a gradient boosting machine. Annals of Statistics pp. 1189–1232 (2001)
14. Hoeffding, W.: Probability inequalities for sums of bounded random variables. In: The Collected Works of Wassily Hoeffding, pp. 409–426. Springer (1994)

15. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.: Lightgbm: A highly efficient gradient boosting decision tree. In: NeurIPS. pp. 3146–3154 (2017)
16. Kosko, B., Burgess, J.C.: Neural networks and fuzzy systems (1998)
17. Kotsiantis, S.B., Pintelas, P.E.: Logitboost of simple bayesian classifier. Informatica (Slovenia) **29**(1), 53–60 (2005)
18. Ledezma, A., Aler, R., Sanchis, A., Borrajo, D.: Empirical evaluation of optimized stacking configurations. In: ICTAI. pp. 49–55. IEEE Computer Society (2004)
19. Liu, A., Lu, J., Zhang, G.: Concept drift detection: Dealing with missing values via fuzzy distance estimations. IEEE Transactions on Fuzzy Systems (2020)
20. Liu, A., Lu, J., Zhang, G.: Concept drift detection via equal intensity k-means space partitioning. IEEE Transactions on Cybernetics (2020)
21. Lu, J., Liu, A., Dong, F., Gu, F., Gama, J., Zhang, G.: Learning under concept drift: A review. IEEE Transactions on Knowledge and Data Engineering **31**(12), 2346–2363 (2018)
22. Mason, L., Baxter, J., Bartlett, P., Frean, M.: Boosting algorithms as gradient descent in function space. In: Proc. NIPS. vol. 12, pp. 512–518 (1999)
23. Montiel, J., Read, J., Bifet, A., Abdessalem, T.: Scikit-multiflow: A multi-output streaming framework. Journal of Machine Learning Research **19**, 72:1–72:5 (2018)
24. Natekin, A., Knoll, A.: Gradient boosting machines, a tutorial. Frontiers in Neurorobotics **7**, 21 (2013)
25. Quinlan, J.R.: Induction of decision trees. Machine Learning **1**(1), 81–106 (1986)
26. Quinlan, J.R.: C4. 5: programs for machine learning. Elsevier (2014)
27. Schapire, R.E.: The strength of weak learnability. Machine Learning **5**(2), 197–227 (1990)
28. Shannon, C.E.: A mathematical theory of communication. The Bell System Technical Journal **27**(3), 379–423 (1948)
29. Umanol, M., Okamoto, H., Hatono, I., Tamura, H., Kawachi, F., Umedzu, S., Kinoshita, J.: Fuzzy decision trees by fuzzy id3 algorithm and its application to diagnosis systems. In: Fuzz-IEEE. pp. 2113–2118. IEEE (1994)
30. Vaishali, R., Sasikala, R., Ramasubbareddy, S., Remya, S., Nalluri, S.: Genetic algorithm based feature selection and moe fuzzy classification algorithm on pima indians diabetes dataset. In: ICCNI. pp. 1–5. IEEE (2017)
31. Vapnik, V.: Principles of risk minimization for learning theory. In: NeurIPS. pp. 831–838 (1992)
32. Wang, K., Lu, J., Liu, A., Zhang, G., Xiong, L.: Evolving gradient boost: A pruning scheme based on loss improvement ratio for learning under concept drift. IEEE Transactions on Cybernetics (2021)
33. Yuan, Y., Shaw, M.J.: Induction of fuzzy decision trees. Fuzzy Sets and Systems **69**(2), 125–139 (1995)
34. Zadeh, L.A.: Fuzzy sets. In: Fuzzy sets, fuzzy logic, and fuzzy systems: selected papers by Lotfi A Zadeh, pp. 394–432. World Scientific (1996)
35. Zhong, L., Fang, Z., Liu, F., Lu, J., Yuan, B., Zhang, G.: How does the combined risk affect the performance of unsupervised domain adaptation approaches? In: AAAI. pp. 11079–11087. AAAI Press (2021)