An Ensemble Tree Classifier for Highly Imbalanced Data Classification*

SHI Peibei · WANG Zhong

DOI: 10.1007/s11424-021-1038-8
Received: 22 February 2021 / Revised: 21 May 2021

Abstract The performance of traditional imbalanced classification algorithms is degraded when dealing with highly imbalanced data. How to deal with highly imbalanced data is a difficult problem. In this paper, the authors propose an ensemble tree classifier for highly imbalanced data classification. The ensemble tree classifier is constructed with a complete binary tree structure. A mathematical model is established based on the features and classification performance of the classifier, and it is proven that the model parameters of the ensemble classifier can be solved by calculation. First, the AdaBoost method is used as the benchmark classifier to construct the tree structure model. Then, the classification cost of the model is calculated, and the quantitative mathematical description between the cost and features of the ensemble tree classifier model is obtained. Then, the cost of the classification model is transformed into an optimization problem, and the parameters of the integrated tree classifier are given through theoretical derivation. This approach is tested on several highly imbalanced datasets in different fields and takes the AUC (area under the curve) and F-measure as evaluation criteria. Compared with the traditional imbalanced classification algorithm, the ensemble tree classifier has better classification performance.

Keywords Ensemble learning, F-measure, imbalanced classification, mathematical model.

1 Introduction

Imbalanced classification refers to an imbalance regarding the number of training samples, such as a large difference between the numbers of positive and negative samples. In daily life, imbalanced classification problems existed in various fields, such as credit card fraud detection[1], text classification[2], information retrieval and filtering[3], market behavior analysis[4], petroleum

SHI Peibei · WANG Zhong (Corresponding author)
School of Computer Science and Technology, Hefei Normal University, Hefei 230601*, China.*
Email: pb_shi@163.com; zhongw@ustc.edu.cn.

*This work was supported by the National Natural Science Foundation of China under Grant No. 61976198, the Natural Science Research Key Project for Colleges and Universities of Anhui Province under Grant No. KJ2019A0726, and the High-level Scientific Research Foundation for the Introduction of Talent of Hefei Normal University under Grant No. 2020RCJJ44.

◇ *This paper was recommended for publication by Editor ZHANG Xinyu.*

surveying[5], medical diagnosis[6], network intrusion detection[7], and software fault detection[8]. For example, in credit card fraud detection problems, the vast majority of transactions are normal transactions; only a few are illegal transactions, and this part is also the focus of researchers.

The solution to the imbalanced classifier problem mainly includes a data level an algorithm level. The data level starts from the given dataset and solves the imbalanced rate of data by resampling the training dataset; such approaches mainly include oversampling technology[9, 10] and undersampling technology[11, 12]. The algorithm level mainly includes cost-sensitive learning[13, 14], one-class learning[15, 16], ensemble classifiers[17, 18] and deep learning[19–22].

Highly imbalanced data usually mean that the positive and negative sample ratios in the dataset are large, generally more than 10. The performance of common imbalanced classification algorithms is degraded when solving highly imbalanced datasets. Figure 1 shows the F-measure of the AdaBoost algorithm on the eighthr dataset with an imbalance ratio from 1:1 to 14:1. The abscissa is the imbalance ratio, and the ordinate is the F-measure index. When the imbalance ratio is 1:1, the F-measure of the AdaBoost algorithm exceeds 0.8. When the imbalance ratio is 14:1, the F-measure of the AdaBoost algorithm is less than 0.45. It can be seen from the figure that with the increase in the imbalance ratio, the F-measure index gradually decreases. Therefore, it is a challenge to study the classification of highly imbalanced data.

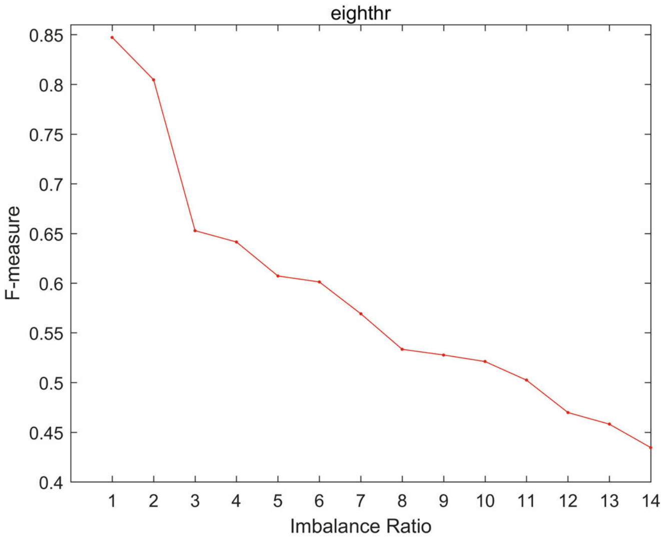

Figure 1 F-measures of the AdaBoost method on the eighthr dataset with different imbalance ratios

There are many highly imbalanced problems in the field of computer vision, such as facial recognition and pedestrian detection. There are typically few samples of faces or pedestrians in the image to be detected, and most of them are negative samples. Sahbi and Geman[23] proposed a hierarchical tree classifier and successfully applied it to facial recognition. Its single classifier is an SVM (support vector machine) classifier, which cannot be directly used for highly imbalanced classification. Inspired by this idea, this paper proposes an ensemble tree classifier model. This method is designed for highly imbalanced datasets, and the purpose is to iteratively divide a difficult, highly imbalanced classification problem into two subproblems. This strategy reduces the classification difficulty of each subproblem. Therefore, we use a complete binary classification tree architecture to build the model. In this method, AdaBoost[24] is used as a single classifier. By analyzing the comprehensive performance of the ensemble tree classifier, the relationship between the performance, features and false alarm rate is obtained, and this relationship is transformed into a quantitative description of the cost of the classification model. Finally, we turn the mathematical model into an optimization problem with constraints and give its parameter solving process.

In summary, the main contributions of this paper are as follows.

- We propose an ensemble tree classifier method, including its mathematical description (EnsembleTree) and minimization process.
- We apply this method to five highly imbalanced datasets in different fields and verify the effectiveness of the proposed method.

The rest of the content in this paper is arranged as follows: Section 2 introduces the related work, and Section 3 introduces the description and mathematical derivation process of the ensemble tree classifier. In Section 4, we test and verify the proposed approach on datasets in different fields. Finally, the summary and future prospects of our research are given.

2 Related Works

The typical oversampling method is the SMOTE approach proposed by Chawla, et al.[9]. This method can increase the number of samples by adding useful information to the training set. The experimental results show that this method is far superior to random oversampling technology. Han, et al.[10] improved the SMOTE method and obtained better classification results than those yielded by the original SMOTE. Zheng, et al.[25] prevented new samples in SMOTE from being limited to the line segment between two seed samples by increasing the number of seeds. Oversampling technology adds a few kinds of data, while undersampling technology removes noise and redundant data. Commonly used undersampling techniques include unilateral selection, editing techniques, consistent subsets, etc.[12]. These methods mainly use heuristic methods and use KNN rules to identify samples that can be removed. Triguero, et al.[26] designed a parallel model to enable evolutionary technology to deal with large-scale classification problems. Due to the defects of both oversampling and undersampling techniques, hybrid sampling techniques have attracted increasing attention[11]. Loyola Gonz'alez used a hybrid sampling technique to improve the accuracy of classifiers based on new patterns.

Cost-sensitive learning fully considers the misclassification costs of different categories. The metacost method proposed by Dominigos[27] estimates the posterior probability density for the given training samples and determines the category of each sample (combined with the cost matrix). Chen, et al.[28] proposed a weighted random forest algorithm, in which the class with the smallest training sample is given the largest weight. Chew, et al.[29] used an SVM to set penalty coefficients for different types of samples by analyzing prior information contained in the training set. Cost-sensitive learning can effectively improve the classification performance of a model for a few classes, but in most cases, it is difficult to accurately estimate the real misclassification cost. Huang and Lin[30] proposed a label embedding strategy that considers the cost function of interest. Lu, et al.[31] embedded the error classification cost, test cost and rejection cost into the rotation forest algorithm to reduce the total classification cost. For cost-sensitive learning, determining the cost is an important problem. One-class learning is also used in imbalanced classification; for example, Raskutti and Kowalczyk[15] proved that one-class learning plays an important role in feature spaces when there are a large number of noisy features. Jusczak and Duin[16] combined a class of learning and resampling techniques to add useful information to the given training set. Ayyagari[32] propose a hybrid of a one-class SVM, k-nearest neighbors and cart algorithms.

The ensemble classifier is a mature technology used to solve imbalanced classification problems. Zhou and Liu[33] proposed a method combining a cost-sensitive neural network and a classifier ensemble. The experimental results on UCI datasets showed that the method is effective for two and multiclass imbalanced problems. Liu, et al.[34] proposed two ensemble classifiers, EasyEnsemble and BalanceCascade, based on the resampling technique. The core idea of EasyEnsemble is to independently extract the same numbers of training sets from the negative sample set and positive sample set each time and then train the AdaBoost classifier with the positive sample set. The final result is the sum of the outputs of all AdaBoost algorithms. The method of sample extraction in BalanceCascade is the same as that of EasyEnsemble. The difference is that the threshold of the AdaBoost classifier is controlled during the training process for each layer of the classifier so that the false positive rate of the classifier is equal to the set value. At the same time, each layer is redivided into the next layer by the negative samples of incorrect classifications to form a new dataset. Compared with AdaBoost, bagging, SMOTE-Boost, AsymBoost, random forests and other traditional ensemble classifiers, EasyEnsemble and BalanceCascade have better classification performance. Galar, et al.[35] reviewed imbalanced classification methods based on ensemble learning, including bagging, boosting and their combination methods, and conducted a large number of tests on UCI datasets. Wang, et al.[36] proposed an undersampling method based on online bagging that can effectively modify the learning bias from the majority to the minority by adaptive weight adjustment. Dubey, et al.[37] proposed an integrated system based on feature selection and data sampling for imbalanced classification.

Deep learning has been successfully applied in computer vision. Most of the existing deep learning methods consider class-balanced data or moderately imbalanced data during model training but ignore the challenge of learning with highly imbalanced training data. Therefore, researchers use a combination of deep learning and category imbalance technology to solve imbalanced classification problems. Jeatrakul, et al.[38] combined SMOTE and a comprehensive neural network to handle imbalanced data classification. Yan, et al.[39] integrated bootstrapping methods and CNN methods for the imbalanced classification of multimedia data. Huang, et al.[40] proposed a deep learning method that combines a simple k-nearest neighbors (KNN) algorithm and can be successfully applied to visual classification tasks. Khan, et al.[41] proposed a cost-sensitive deep network that can automatically learn robust feature representations for both the majority and minority classes. Dong, et al.[42] proposed a class-imbalanced deep learning model with a class rectification loss function.

3 Ensemble Tree Classifier

3.1 Problem Statement

EnsembleTree is a hierarchical classifier that is described by a complete binary tree structure. For highly imbalanced classification problems, the numbers of positive and negative samples vary greatly. The ensemble tree method adopts the strategy of early screening, and the overall classification cost is small. For a positive sample, a complete path from the root node to the leaf node is needed, while for a negative sample, access to the classifier is less important and the cost is lower. Figure 2 shows the schematic diagram of the complete binary classification tree, in which black circles represent positive samples and white circles represent negative samples. Because of the complete binary tree structure of the ensemble tree, the depth-first search method is used for its classification search process.

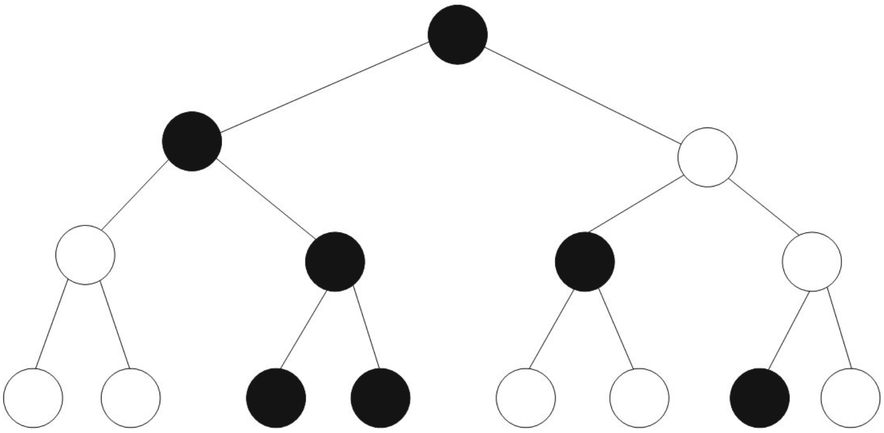

Figure 2 The structure of EnsembleTree

For a single-node classifier, the AdaBoost classifier is used. Suppose that for an AdaBoost classifier with n features, the cost of classification is cost= $a \times n + b$, where a and b are parameters. Suppose that the complete binary tree has L layers, $C_{l,k}$ denotes the kth classifier of layer l, $n_{l,k}$ represents the number of features used by the kth classifier of the lth layer, and v_l is the number of nodes with layer number l; then, $v_l = 2^{l-1}$. Considering that AdaBoost uses the same number of features in each layer of the tree structure, $n_{l,k} = n_l$.

Suppose that $\delta(l-1;n)$ indicates the probability that a sample is classified incorrectly by layer $1, 2, \cdots, l-1$ and that $\delta(0;n) = 1$; then, the classification cost of a sample is

$$\text{cost } = \sum_{l=1}^{L} \sum_{k=1}^{v_l} 1_{\{C_{l,k} \text{ is performed}\}} n_{l,k}.$$

Since this paper considers a highly imbalanced classification problem, the numbers of positive and negative samples are quite different. Compared with that of negative samples, we assume that the classification cost of positive sample objects is negligible, and the overall cost of all ensemble tree classifiers mainly considers the classification cost of negative samples. For negative samples, if a negative sample object is detected in the lth layer, then the probability of its execution is equivalent to the probability that the sample is incorrectly judged as "positive" by the $1, 2, \cdots, l-1$ layer classifier (false positive rate). Therefore, the classification cost of the ensemble tree classifier is shown in Formula (1):

$$\begin{aligned}\mathrm{E(cost)} &= \sum_{l=1}^{L}\sum_{k=1}^{v_l} 1_{\{C_{l,k} \text{ is performed}\}} n_{l,k} \\ &= \sum_{l=1}^{L}\sum_{k=1}^{v_l} \delta(l-1;n) n_{l,k} \\ &= \sum_{l=1}^{L} v_l \delta(l-1;n) n_l \\ &= n_1 + \sum_{l=2}^{L} v_l n_l \delta(l-1;n). \end{aligned} \tag{1}$$

The false positive rate of the entire ensemble tree classifier is the sum of the false positive rates of all single classifiers in the Lth layer $v_L\delta(L;n)$. To ensure the optimal performance of the model on an imbalanced classification problem, the classification cost must be minimized, and the above cost can finally be converted into a minimization problem.

$$\begin{aligned} &\min_{n_1,n_2,\cdots,n_L} n_1 + \sum_{l=2}^{L} v_l n_l \delta(l-1;n) \\ &\text{s.t.} \begin{cases} v_L \delta(L;n) \le \mu, \\ 0 < n_l \le N_l, \end{cases} \end{aligned} \tag{2}$$

where μ represents the upper limit of the number of false positive rates and N_l represents the maximum number of features that can be used by the lth layer of a single classifier. According to the marginal and conditional probabilities, the functional relationship between $\delta(l-1;n)$, l and n can be further derived. Assuming that

$$\delta(l-1;n) = \left(\sum_{j=1}^{l} \beta_j n_j\right)^{-1},$$

where $\beta_1, \beta_2, \cdots, \beta_l$ are constants, the minimization problem in Formula (2) can be transformed as follows:

$$\begin{aligned}
&\min_{n_1, n_2, \cdots, n_L} n_1 + \sum_{l=2}^{L} v_l n_l \left(\sum_{j=1}^{l} \beta_j n_j \right)^{-1} \\
&\text{s.t.} \begin{cases} v_L \left(\sum_{j=1}^{L} \beta_j n_j \right)^{-1} \leq \mu, \\ 0 < n_l \leq N_l. \end{cases}
\end{aligned} \tag{3}$$

3.2 Optimal Solution

To minimize the optimization time of the ensemble tree classifier method, this section gives the specific process for solving the above formula. Our goal is to obtain a set of feature values $n_1, n_2, \cdots, n_l$ that minimizes the classification cost of the classification model. Assuming that

$$C = \min_{n_1, n_2, \cdots, n_L} n_1 + \sum_{l=2}^{L} v_l n_l \left(\sum_{j=1}^{l} \beta_j n_j \right)^{-1},$$

when n_L is determined, the specific solving process is as follows. First, the partial derivatives of C with respect to n_1 and n_j are obtained, as shown in Formula (4).

$$\begin{aligned}
&\frac{\partial C}{\partial n_1} = 1 - \sum_{l=2}^{L} \left[\frac{\beta_1 2^{l-1} n_l}{\left(\sum_{i=1}^{l-1} \beta_i n_i \right)^2} \right], \\
&\frac{\partial C}{\partial n_j} = \frac{2^{j-1}}{\sum_{i=1}^{j-1} \beta_i n_i} - \sum_{l=j+1}^{L} \left[\frac{\beta_j 2^{l-1} n_l}{\left(\sum_{i=1}^{l-1} \beta_i n_i \right)^2} \right], \quad j \in \{2, L-1\}.
\end{aligned} \tag{4}$$

Furthermore,

$$\begin{aligned}
&\frac{\partial C}{\partial n_{j+1}} = 0 \Rightarrow \sum_{l=j+2}^{L} \left[\frac{2^{l-1} n_l}{\left(\sum_{i=1}^{l-1} \beta_i n_i \right)^2} \right] = \frac{2^j}{\sum_{i=1}^{j-1} \beta_i n_i} \frac{1}{\beta_{j+1}}, \\
&\frac{\partial C}{\partial n_j} = 0 \Rightarrow \frac{2^{j-1}}{\sum_{i=1}^{j-1} \beta_i n_i} - \beta_j \frac{2^j n_{j+1}}{\left(\sum_{i=1}^{j} \beta_i n_i \right)^2} = \beta_j \sum_{l=j+2}^{L} \left[\frac{2^{l-1} n_l}{\left(\sum_{i=1}^{l-1} \beta_i n_i \right)^2} \right].
\end{aligned} \tag{5}$$

According to Formula (5), we can obtain:

$$\frac{2^{j-1}}{\sum_{i=1}^{j-1} \beta_i n_i} - \beta_j \frac{2^j n_{j+1}}{\left(\sum_{i=1}^{j} \beta_i n_i \right)^2} - \beta_j \frac{2^j}{\beta_{j+1} \left(\sum_{i=1}^{j} \beta_i n_i \right)} = 0, \quad j \neq 1. \tag{6}$$

Assuming that n_1 is known, according to $\frac{\partial C}{\partial n_1} = 0$ and $\frac{\partial C}{\partial n_2} = 0$, we can obtain:

$$1 - \frac{\beta_1 2 n_2}{\beta_1^2 n_1^2} - \frac{2\beta_1}{\beta_2 \beta_1 n_1} = 0 \Rightarrow n_2 = \frac{1}{2} \frac{\beta_1}{\beta_2} n_1 \left(\beta_2 n_1 - 2 \right). \tag{7}$$

Assuming that $2 \leq j \leq l$ ($l \in \{2, L-2\}$),

$$n_j = 2^{-\frac{1}{2}j(j-1)} \left(\prod_{i=1}^{j-1} \beta_i \right) \beta_j^{-1} n_1^{j-1} \left(\beta_j n_1 - 2^{j-1} \right). \tag{8}$$

Now, we need to prove Formula (9):

$$n_{l+1} = 2^{-\frac{1}{2}l(l+1)} \left(\prod_{i=1}^{l} \beta_i \right) \beta_{l+1}^{-1} n_1^{l} \left(\beta_{l+1} n_1 - 2^{l} \right). \tag{9}$$

According to Formula (8), for any $j \in \{2, l\}$ we have

$$\begin{aligned} \sum_{i=1}^{j} \beta_i n_i = {} & \beta_1 n_1 + \beta_2 \frac{1}{2} \beta_1 \beta_2^{-1} n_1 + \beta_3 \frac{1}{8} \beta_1 \beta_2 \beta_3^{-1} n_1^2 \left(\beta_3 n_1 - 4 \right) + \cdots \\ & + \beta_{j-1} \left(2^{-\frac{1}{2}(j-1)(j-2)} \right) \left(\prod_{i=1}^{j-2} \beta_i \right) \beta_{j-1}^{-1} n_1^{j-2} \left(\beta_{j-1} n_1 - 2^{j-2} \right) \\ & + \beta_j \left(2^{-\frac{1}{2}j(j-1)} \right) \left(\prod_{i=1}^{j-1} \beta_i \right) \beta_j^{-1} n_1^{j-1} \left(\beta_j n_1 - 2^{j-1} \right). \end{aligned} \tag{10}$$

Therefore, for any $j \in \{2, l\}$,

$$\sum_{i=1}^{j} \beta_i n_i = 2^{-\frac{1}{2}j(j-1)} \left(\prod_{i=1}^{j-1} \beta_i \right) n_1^{j}. \tag{11}$$

Assuming that $\pi_j = \prod_{i=1}^{j} \beta_i$, by substituting $j = l$ into Formula (10), we can rewrite Formula (6).

$$\begin{aligned} & \frac{2^{l-1}}{2^{-\frac{1}{2}(l-1)(l-2)} \pi_{l-1} n_1^{l-1}} - \beta_l \frac{2^l n_{l+1}}{\left(2^{-\frac{1}{2}(l-1)} \pi_l n_1^l \right)^2} \\ & - \frac{\beta_l}{\beta_{l+1}} \frac{2^l}{2^{-\frac{1}{2}(l-1)} \pi_l n_1^l} = 0 \Rightarrow \\ & n_{l+1} = 2^{-\frac{1}{2}(l+1)} \pi_l \beta_{l+1}^{-1} n_1^l \left(\beta_{l+1} n_1 - 2^l \right). \end{aligned} \tag{12}$$

Thus, we prove Formula (9). For n_1, by substituting $j = L-1$ into Formula (4), we can obtain:

$$\frac{\partial C}{\partial n_{L-1}} = 0 \Rightarrow n_1 = \left(\frac{1}{\pi_{L-1}} 2^{L(L-1)/2} n_L \right)^{1/L}. \tag{13}$$

We rewrite Formula (3) to acquire:

$$\begin{aligned} & \min_{n_L} L \left(\frac{1}{\pi_{L-1}} 2^{L(L-1)/2} n_L \right)^{1/L} - \sum_{l=2}^{L} \left(\frac{2^{l-1}}{\beta_l} \right) \\ & \text{s.t.} \begin{cases} v_L \left(\sum_{j=1}^{L} \beta_j n_j \right)^{-1} \leq \mu, \\ 0 < n_l \leq N_l. \end{cases} \end{aligned} \tag{14}$$

Formula (14) is merely a function of n_L. Finally, we provide the solution result of Formula (3).

$$n_{l=} \begin{cases} \left(2^{L(L-1)/2}\left(\prod_{i=1}^{L-1}\beta_i\right)^{-1} n_L\right)^{1/L}, & l=1, \\ 2^{-l(l-1)/2}\left(\prod_{i=1}^{l-1}\beta_i\right)\beta_l^{-1} n_1^{l-1}\left(\beta_l n_1 - 2^{l-1}\right), & l\in\{2,3,\cdots,L-1\}, \\ n_L, & l=L. \end{cases} \tag{15}$$

After summarizing the above mathematical model and the process of minimizing the proof of the solution, the ensemble tree classifier model is proposed for problems with highly imbalanced data, and the parameter solving procedure can be transformed into a constrained optimization problem. The two parameters of the constraint are the false positive rate and the upper limit of the number of features used by the lth layer of a single classifier. The false positive rate can be manually limited, and the number of features can be automatically obtained according to the given dataset.

The ensemble tree classifier model optimizes the classification cost under the premise of meeting the preset detection rate and false positive rate. Its complete binary tree structure can effectively solve problems with highly imbalanced training samples and can divide complex classification problems into two reduced-difficulty subproblems. Regarding the classification performance, the false positive rate of the ensemble tree classifier model can be manually set, and the detection rate can be satisfied by setting the detection rate of a single classifier, so the ensemble tree classifier model can improve upon the classification performance of other methods.

3.3 Algorithm

Algorithm 1 provides the specific training process of EnsembleTree. The single feature classifier (AdaBoost) uses quicksort to sort and search for determining the theoretically optimal threshold of each column of feature values under the current weight. During sample selection, each node selects a subset from the current negative sample set with the same number of samples as that in the positive sample set and uses these negative samples and all positive samples for training. The negative samples that are misjudged enter the next layer, and the positive samples are not split. When the number of features is sufficient during feature selection, each node uses only the features that have not been used by ancestor nodes and sibling nodes to construct a single feature classifier. When the number of features is insufficient, a single feature classifier is selected and constructed from the features used by ancestor nodes or sibling nodes. Considering the complexity of EnsembleTree, we manually set the binary tree to have 3 levels, so we only need to linearly fit the values β_1 and β_2.

Algorithm 1 EnsembleTree training process

```
Input imbalanced data, including the number of features and the false positive rate $\mu$
Output EnsembleTree
Perform linear regression according to the least-squares method using validation data, and
calculate $\beta_1$, $\beta_2$ by fitting algorithm
Use Formula (15) to calculate the number of features in each layer $n_1, n_2, n_3$
for iterations =1 to ROUND do
    Randomly shuffle the original dataset
    for $j = 1$ to CROSS_FOLD do
        Extract a training subset
        Recursively train the left and right subtrees of the binary tree to obtain EnsembleTree
        Use EnsembleTree to test other subsets and calculate the AUCs and F-measures
    end for
end for
```

4 Experiments

4.1 Dataset

Five datasets from different fields are selected, including pedestrian feature data, facial feature data and UCI datasets. Among them, eighthr is data regarding the surface ozone level during an 8-hour peak period, OBP contains odorant binding protein gene data, FICCBBR is a multiclass gene sequence dataset that includes seven categories, and the pedestrian and facial datasets contain image data with a small number of pedestrians and faces, where most of their backgrounds are nontarget objects. Pedestrian and facial features are obtained by extracting Haar-like features from the corresponding images[24]. Since Haar-like features are massive, a feature optimization algorithm based on coevolution is used to select the top 5000 features[43]. Table 1 shows the specific descriptive information of the different datasets. Table 1 shows that all datasets contain highly imbalanced data, and their IRs range from 11 to 55.

Table 1 Descriptions of the test datasets

Dataset	Number of features	Number of positive samples	Number of negative samples	Imbalance Rate
eighthr	72	160	2374	14.838
OBP	1463	108	2157	19.972
FICBBRC	461	678	37854	55.832
Pedestrian	5000	400	4500	11.25
Facial	5000	400	6227	15.56

4.2 Evaluation Measures

Common indicators for imbalanced classification problems are the ROC (receiver operating characteristic curve) and the AUC (area covered under the ROC curve)[44]. These evaluation indicators are based on a confusion matrix, as shown in Table 2.

Table 2 Confusion matrix

Classes	Predictive minority class	Predictive majority class
Actual minority class	True Positives (TPs)	False Negatives (FNs)
Actual majority class	False Positives (FPs)	True Negatives (TNs)

According to the confusion matrix, the commonly used evaluation indices for imbalanced classification problems can be obtained as follows.

Precision: $P = TP/(TP + FP)$.

Recall: $R = TP/(TP + FN)$.

F-measure= $2PR/(P + R)$.

The F-measure is the harmonic average of recall and precision, and its value is closer to the smaller of the two. A high F-measure value can ensure high recall and precision. As a reliable evaluation standard, the AUC is suitable for imbalanced classification and cost-sensitive problems. When comparing the positive and negative sample scores, if a positive sample score is higher than the corresponding negative sample score, 1 point is accumulated; if the positive sample score is equal to the negative sample score, 0.5 points are accumulated; if the positive sample score is less than the negative sample score, 0 points are accumulated.

4.3 Algorithm Comparison and Analysis

Each group of data is used for 5-fold cross validation (four are randomly selected as the training sets, and one is used as the test set), and the average results of 20 runs are statistically averaged. Therefore, in Algorithm 1, ROUND is set to 20 and CROSS_FOLD is 5. We compare the following methods:

1) AdaBoost (Ada): The following algorithms are based on the AdaBoost algorithm.

2) UnderSampling+AdaBoost (Under): This method randomly selects a subset of the negative sample set with the same number of samples as that in the positive set and uses the positive sample set and the negative sample subset to train the AdaBoost classifier.

3) EasyEnsemble (Easy): A subset of the negative sample set with the same number of samples as that in the positive sample set is independently extracted, and the AdaBoost classifier is trained with the positive sample set and the negative sample subset. The final discriminant function is the superposition of each AdaBoost classifier, and the number of layers in the experiment is 4.

4) BalanceCascade (Balance): A subsets of the negative sample set with the same number of samples as that in the positive sample set is independently extracted, the AdaBoost classifier is trained with the positive sample set and the negative sample subset, and the threshold of

each layer of the AdaBoost classifier is controlled to make the false positive rate equal to the hypothetical value, which finally makes the positive and negative sample sizes of the last layer equal; the experimental fixed number of layers is 4.

5) EnsembleTree (Ensemble): The number of layers in the experiment is 3 layers. Table 3 shows the false positive rate, average features, number of features in each layer n_i and the corresponding parameter value β_i of the EnsembleTree method on each of the five test datasets.

6) OKC classifier (OKC): This is a hybrid combination of the one-class SVM, KNN and CART algorithms with default parameters.

Table 3 The parameters of the EnsembleTree method

Dataset	False positive rate	Average features	n_1	n_2	n_3	β_1	β_2
eighthr	0.08	22.7	13	14	30	0.3798	0.2685
OBP	0.062	58.5	20	29	60	0.3506	0.1711
FICBBRC	0.15	24.2	11	14	30	0.7783	0.2611
Pedestrian	0.002	40.6	18	27	70	0.2221	0.4107
Face	0.005	65.6	28	35	70	0.1594	0.1618

Considering the complexity of each dataset and the difficulty of classification, the false positive rate is set according to manual experience to achieve the optimal classification performance. Figure 3 shows the F-measure curve corresponding to different false positive rates on the eighthr dataset. The final false positive rate is selected based on the optimal classification result.

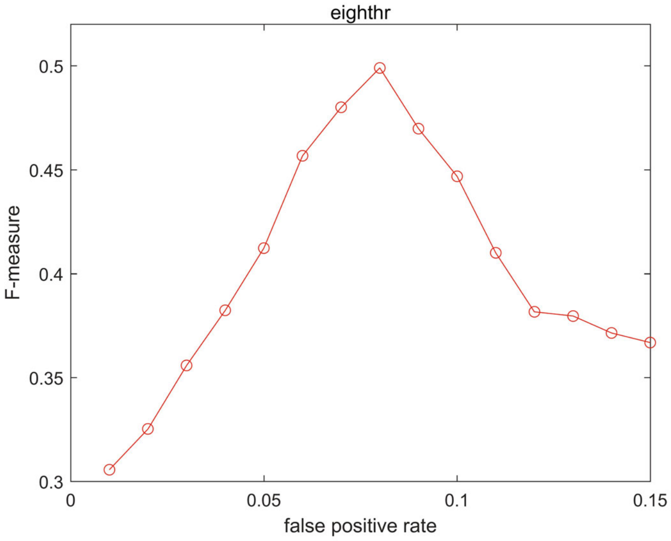

Figure 3 F-measure curve for different false positive rates on the eighthr dataset

Table 4 and Table 5 give the statistical results regarding the AUCs and F-measures of the different algorithms, respectively. It can be seen from the statistical results of Table 4 and

Table 5 that the classification difficulties of these five test datasets are different. The FICBBRC dataset is very difficult, and the F-measure results on it are low (less than 0.2). The difficulty of the PDS dataset is low, and its AUC and F-measure results are both high (higher than 0.9). The other three datasets, eighthr, the facial dataset, and OBP are of moderate difficulty, and their difficulty decreases in the order they are listed above. For different test datasets, each algorithm ranks differently. For example, the F-measure results on the facial dataset are ranked as follows: Ensemble, Cascade, Ada, OKC, Easy, and Under. The common feature of Table 4 and Table 5 is that the results in terms of the AUC and F-measure achieved by the Ensemble method are both optimal, and the advantages of this approach are more obvious when examining the F-measure results. Due to the problem of early rejection in the Ensemble, the scores of some samples are still not further subdivided with subsequent features, so the AUC value advantage is not very obvious.

Table 4 AUC results of different algorithms

Dataset	Ensemble	Ada	Under	Easy	Cascade	OKC
eighthr	**0.8910±0.0272**	0.8862±0.0238	0.8839±0.0237	0.8717±0.0283	0.8818±0.0286	0.8825±0.0326
OBP	**0.9272±0.0238**	0.9246±0.0249	0.9070±0.0259	0.9095±0.0267	0.9014±0.0389	0.9052±0.0368
FICBBRC	**0.8196±0.0162**	0.8096±0.0159	0.8039±0.0174	0.8145±0.0171	0.7663±0.0581	0.7824±0.0164
Pedestrian	**0.9885±0.0070**	0.9803±0.0033	0.9861±0.0052	0.9835±0.0055	0.9872±0.0042	0.9812±0.0012
Face	**0.9403±0.0110**	0.9397±0.0096	0.9281±0.0123	0.9310±0.0107	0.9380±0.0106	0.9325±0.0142

Table 5 F-measure results of different algorithms

Dataset	Ensemble	Ada	Under	Easy	Cascade	OKC
eighthr	**0.4990±0.0650**	0.4771±0.0566	0.4504±0.0531	0.4335±0.0595	0.4607±0.0592	0.4825±0.0834
OBP	**0.7579±0.0702**	0.7403±0.0654	0.6466±0.0902	0.6878±0.0674	0.7123±0.0736	0.7069±0.0392
FICBBRC	**0.1967±0.0234**	0.1823±0.0203	0.1633±0.0233	0.1700±0.0199	0.1382±0.0380	0.1565±0.0262
Pedestrian	**0.9021±0.0241**	0.8498±0.0270	0.8248±0.0362	0.8094±0.0347	0.8368±0.0306	0.8826±0.0748
Face	**0.5835±0.0424**	0.5630±0.0396	0.5128±0.0422	0.5265±0.0368	0.5647±0.0436	0.5527±0.0365

Figure 4 shows the F-measure curves of each method on the five test datasets, where the abscissa is the number of features and the ordinate is the statistical result. The Ensemble method uses average features. Compared with the other four methods, when using the same number of features, the method proposed in the paper can achieve the best results, and the F-measure result of the Ensemble method has obvious advantages. In addition, it can be seen from the statistical curves of the eighthr and pedestrian datasets that the F-measure value of the Ensemble method is better than the best results of the other four methods.

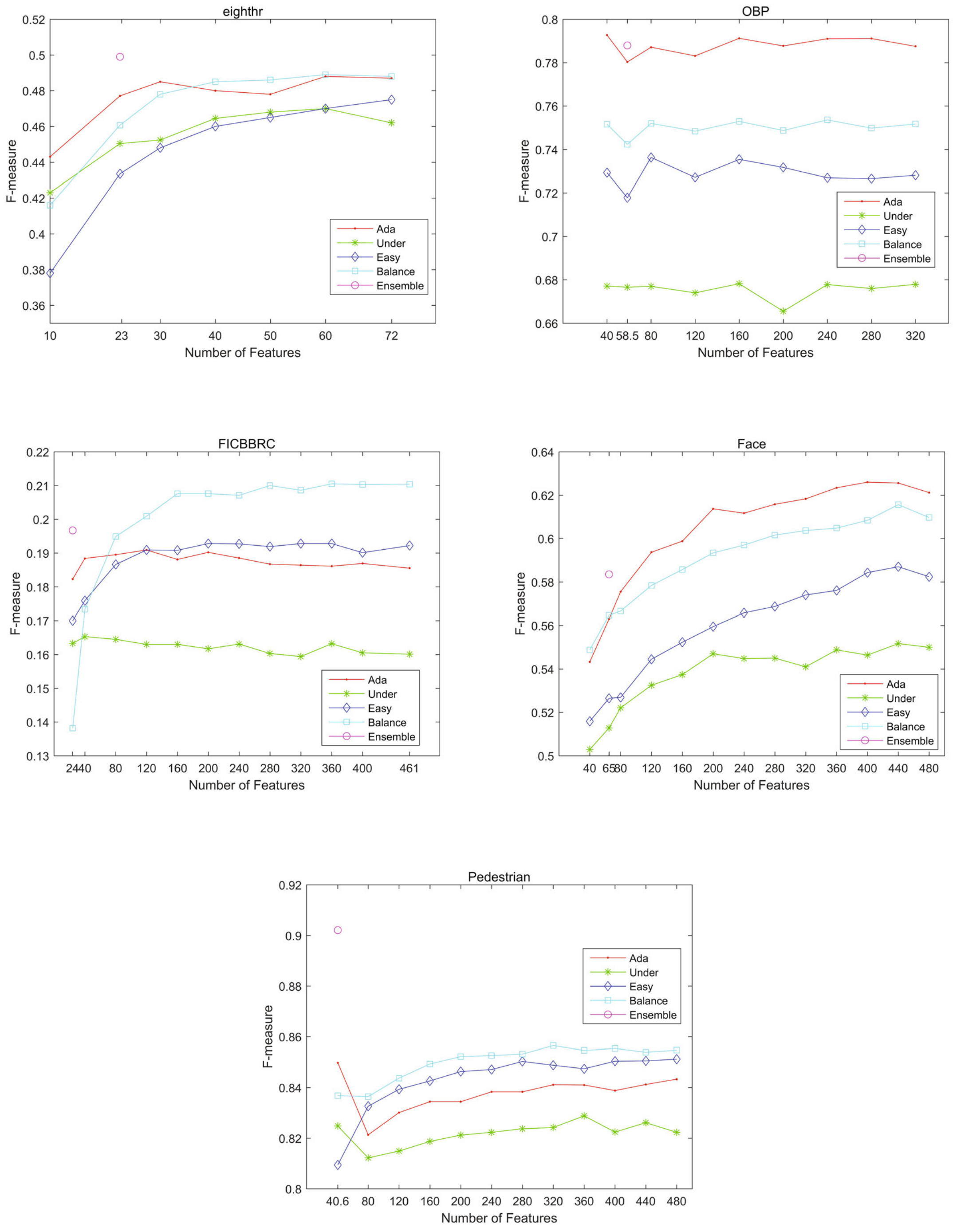

Figure 4 F-measure curves on different test datasets

5 Conclusion

In this paper, an ensemble tree classifier is proposed; it uses a complete binary tree structure to build a model and minimize the model parameters. Experiments are carried out on five highly imbalanced test datasets from different fields. The experimental results show that the ensemble tree classifier has better classification results than those of the competing approaches in terms of the AUC and F-measure. Here, the complete binary tree classification architecture is considered, and the binary tree is solved with the aim of solving binary classification problems. For highly imbalanced multiclass problems, the proposed method has the same practicability. In addition, studying the structures and theoretical solutions of k-ary complete trees are future research directions.

References

[1] Wang X M, Hu M, Zhao Y L, et al., Credit scoring based on the set-valued identification method, *Journal of Systems Science and Complexity*, 2020, **33**(5): 1297–1309.

[2] Sun A X, Lim E P, and Liu Y, On strategies for imbalanced text classification using SVM: A comparative study, *Decision Support Systems*, 2009, **48**(1): 191–201.

[3] Xie L, Jia Y L, Xiao J, et al., GMDH-based outlier detection model in classification problems, *Journal of Systems Science and Complexity*, 2020, **33**(5): 1516–1532.

[4] Burez J and Poel D V D, Handling class imbalance in customer churn prediction, *Expert Systems with Applications*, 2008, **36**(3): 4626–4636.

[5] Brekke C and Solberg A H S, Oil spill detection by satellite remote sensing, *Remote Sensing of Environment*, 2005, **95**(1): 1–13.

[6] Plant C, Bhm C, Tilg B, et al., Enhancing instance-based classification with local density: A new algorithm for classifying unbalanced biomedical data, *Bioinformatics*, 2006, **22**(8): 981–988.

[7] Chen J D and Tang X J, The distributed representation for societal risk classification toward BBS posts, *Journal of Systems Science and Complexity*, 2017, **30**(3): 113–130.

[8] Song Q B, Guo Y C, and Shepperd M, A comprehensive investigation of the role of imbalanced learning for software defect prediction, *IEEE Transactions on Software Engineering*, 2019, **45**(12): 1253–1269.

[9] Chawla N V, Bowyer K W, Hall L O, et al., SMOTE: Synthetic minority over-sampling technique, *Journal of Artificial Intelligence Research*, 2002, **16**(1): 321–357.

[10] Hui H, Wang W Y, and Mao B H, Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning, *Proceedings of the* 2005 *International Conference on Advances in Intelligent Computing*, 2005, 878–887.

[11] Loyola-Gonzlez O, Garca-Borroto M, Medina-Prez M A, et al., An empirical study of oversampling and undersampling methods for LCMine an emerging pattern based classifier, *Mexican Conference on Pattern Recognition*, 2013, 264–273.

[12] Batista G E, Prati R C, and Monard M C, A study of the behavior of several methods for balancing machine learning training data, *ACM SIGKDD Explorations Newsletter*, 2004, **6**(1): 20–29.

[13] Castro C L and Braga A P, Novel cost-sensitive approach to improve the multilayer perceptron performance on imbalanced data, *IEEE Transactions on Neural Networks and Learning Systems*, 2013, **24**(6): 888–899.

[14] Thai-Nghe N, Gantner Z, and Schmidt-Thieme L, Cost-sensitive learning methods for imbalanced data, *International Joint Conference on Neural Networks*, 2010, 1–8.

[15] Raskutti B and Kowalczyk A, Extreme re-balancing for SVMs: A case study, *ACM Sigkdd Explorations Newsletter*, 2004, **6**(1): 60–69.

[16] Juszczak P and Duin R P W, Uncertainty sampling methods for one-class classifiers, *Proceedings of ICML*-03*, Workshop on Learning with Imbalanced Data Sets II*, 2003, 81–88.

[17] Chen Z, Duan J, Kang L, et al., A hybrid data-level ensemble to enable learning from highly imbalanced dataset, *Information Sciences*, 2021, **554**: 157–176.

[18] Yang P Y, Yoo P D, Fernando J, et al., Sample subset optimization techniques for imbalanced and ensemble learning problems in bioinformatics Applications, *IEEE Transactions on Cybernetics*, 2014, **44**(3): 445–455.

[19] Ando S and Huang C Y, Deep over-sampling framework for classifying imbalanced data, *ECML PKDD*, 2017, 770–785.

[20] Zhang C, Tan K C, and Ren R, Training cost-sensitive deep belief networks on imbalance data problems, *International Joint Conference on Neural Networks*, 2016, 4362–4367.

[21] Hu J L, Lu J W, Tan Y P, et al., Deep transfer metric learning, *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, 325–333.

[22] Dong Q, Gong S G, and Zhu X T, Class rectification hard mining for imbalanced deep learning, *IEEE International Conference on Computer Vision*, 2017, 1869–1878.

[23] Sahbi H and Geman D, A hierarchy of support vector machines for pattern detection, *The Journal of Machine Learning Research*, 2006, **7**: 2087–2123.

[24] Viola P and Jones M, Robust real-time object detection, *International Journal of Computer Vision*, 2003, **57**(2): 137–154.

[25] Zheng Z Y, Cai Y P, and Li Y, Oversampling method for imbalanced classification, *Computing and Informatics*, 2016, **34**(5): 1017–1037.

[26] Triguero I, Galar M, Vluymans S, et al., Evolutionary undersampling for imbalanced big data classification, *IEEE Congress on Evolutionary Computation* (*CEC*), 2015, 715–722.

[27] Domingos P, MetaCost: A general method for making classifiers cost-sensitive, *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1999, 155–164.

[28] Chen C, Liaw A, and Breiman L, Using random forest to learn imbalanced data, *No.* 666*, Statistics Department, University of California at Berkeley*, 2004.

[29] Chew H G, Bogner R E, and Lim C C, Dual v-support vector machine with error rate and training size biasing, *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2001, **2**: 1269–1272.

[30] Huang K H and Lin H T, Cost-sensitive label embedding for multi-label classification, *Machine Learning*, 2017, **106**(9–10): 1725–1746.

[31] Lu H J, Yang L, Yan K, et al., A cost-sensitive rotation forest algorithm for gene expression data classification, *Neurocomputing*, 2016, **228**: 270–276.

[32] Ayyagari M R, Classification of imbalanced datasets using one-class SVM, k-nearest neighbors and CART algorithm, *International Journal of Advanced Computer Science and Applications*,

2020, **11**(11): 1–5.

[33] Zhou Z H and Liu X Y, Training cost-sensitive neural networks with methods addressing the class imbalance problem, *IEEE Transactions on Knowledge and Data Engineering*, 2006, **18**(1): 63–77.

[34] Liu X Y, Wu J, and Zhou Z H, Exploratory undersampling for class-imbalance learning, *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 2009, **39**(2): 539–550.

[35] Galar M, Fernandez A, Barrenechea E, et al., A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches, *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 2012, **42**(4): 463–484.

[36] Wang S, Minku L L, and Yao X, Resampling-based ensemble methods for online class imbalance learning, *IEEE Transactions on Knowledge and Data Engineering*, 2015, **27**(5): 1356–1368.

[37] Dubey R, Zhou J Y, Wang Y L, et al., Analysis of sampling techniques for imbalanced data: An $n = 648$ ADNI study, *Neuroimage*, 2014, **87**: 220–241.

[38] Jeatrakul P, Wong K W, and Fung C C, Classification of imbalanced data by combining the complementary neural network and SMOTE algorithm, 17*th International Conference on Neural Information Processing*, 2010, 152–159.

[39] Yan Y L, Chen M, Shyu M L, et al., Deep learning for imbalanced multimedia data classification, *IEEE International Symposium on Multimedia*, 2016, 483–488.

[40] Huang C, Li Y N, Loy C C, et al., Learning deep representation for imbalanced classification, *IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), 2016, 5375–5384.

[41] Khan S H, Hayat M, Bennamoun M, et al., Cost sensitive learning of deep feature representations from imbalanced data, *IEEE Transactions on Neural Networks and Learning Systems*, 2018, **29**(8): 3573–3587.

[42] Dong Q, Gong S G, and Zhu X T, Imbalanced deep learning by minority class incremental Rectification, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019, **41**(6): 1367–1381.

[43] Cao X B, Qiao H, and Keane J, A low-cost pedestrian-detection system with a single optical camera, *IEEE Transactions on Intelligent Transportation Systems*, 2008, **9**(1): 58–67.

[44] Liu X Y, Li Q Q, and Zhou Z H, Learning imbalanced multi-class data with optimal dichotomy weights, *International Conference on Data Mining*, 2013, 478–487.