Available online at www.ijpe-online.com

International Journal of Performability Engineering

vol. 14, no. 6, June 2018, pp. 1140-1148
DOI: 10.23940/ijpe.18.06.p5.11401148

A Novel Imbalanced Classification Method based on Decision Tree and Bagging

Hongjiao Guan[a,*], Yingtao Zhang[a], Hengda Cheng[b], and Xianglong Tang[a]

[a]*School of Computer Science and Technology, Harbin Institute of Technology, Harbin, 150001, China*
[b]*School of Computer Science and Technology, Utah State University, Logan, 84322, USA*

Abstract

Imbalanced classification is a challenging problem in the field of big data research and applications. Complex data distributions, such as small disjuncts and overlapping classes, make traditional methods unable to easily recognize the minority class and thus, lead to low sensitivity. The misclassification costs of the minority class are usually higher than that of the majority class. To deal with imbalanced datasets, typical algorithmic-level methods either introduce cost information or simply rebalance class distribution without considering the distribution of the minority class. In this paper, we propose an optimization embedded bagging (OEBag) approach to increase the sensitivity by learning the complex distributions in the minority class more precisely. By learning these base classifiers, OEBag selectively learns the minority examples that are misclassified easily by referring to examples in out-of-bag. OEBag is implemented by using two specialized under-sampling bagging methods. Nineteen real datasets with diverse levels of classification difficulties are utilized in this paper. Experimental results demonstrate that OEBag performs significantly better in sensitivity and has a great overall performance in terms of AUC (area under ROC curve) and G-mean when compared with several state-of-the-art methods.

Keywords: imbalanced classification; decision tree; bagging; overlapping among classes; small disjuncts

(Submitted on March 6, 2018; Revised on April 16, 2018; Accepted on May 21, 2018)

1. Introduction

Imbalanced datasets are very common in the field of big data and real applications, especially, in safety critical areas, such as medical diagnosis [22], intrusion detection [21], etc. The class with less examples is usually called the positive class or the minority class; the class with more examples is usually called the negative class or the majority class. Typically, traditional classification methods are not suitable for such datasets, since they often assume approximately balanced class distribution or equal misclassification cost, and they try to minimize overall error rate or maximize accuracy. When confronting imbalanced datasets, the algorithms are often in favor of the majority class [5]. Consequently, decision boundary inclines to the minority class space, leading to many misclassifications of the minority class examples. Imbalanced recognition rates also attribute to the imbalanced dataset itself, such as small sizes of the positive class and complex data distributions [11]. Small sizes of the positive class indicate the absolute imbalance or imbalance due to rarity. Rarity leads to inadequate sub-concepts, and learning methods do not have sufficient training samples or well-represented class distributions to deduce the appropriate boundary.

Complex data distributions are key factors that lead to imbalanced recognition rates of the minority class and the majority class. Note that in the paper we only consider the binary imbalanced classification. Complex distributions mainly contain small disjuncts [10], overlapping classes [16], and too many rare cases and outliers [14] in the minority class space. Small disjuncts indicate the situation where a homogeneous area has several small clusters with different class labels. Basically, small disjuncts occur in the minority class space as a direct result of underrepresented cases. Decision trees tend to either specialize rules that cover disjunct examples or ignore these examples. Overlapping classes occurs near the class borders, where some instances from different classes have similar attributes. These borderline examples can obscure the class boundary

* Corresponding author.
E-mail address: guanhongjiao2008@163.com

and increase the likelihood of misclassifying minority class examples. Rare cases and outliers are the result of the underrepresented minority class, which are much likely to be identified as noisy examples. Usually, these difficult factors do not occur alone [19].

On the other hand, correct recognition of the minority class is of crucial implication from the practical application viewpoint. Compared with the majority class, the minority class represents a more important aspect that we are interested in. The minority class is usually associated with a higher misclassification risk. For example, the correct recognition of cancer cases is more worthwhile than that of normal cases, since it is related to high cost and even a life risk. In light of these opinions, we can conclude that there exists a paradox between the inevitability of classification difficulty and the desirability of high recognition rate of the minority class in imbalanced datasets. Notice that we stress the high identification importance of the minority class; however, it does not mean that the recognition of the majority class is not important. We aim to improve the sensitivity of the minority class and ensure the overall performance at the same time.

Many works have studied the imbalanced classification problem and developed preprocessing techniques, cost-sensitive learning or ensemble approaches [11,12]. Preprocessing includes over-sampling and under-sampling, which modify the class distribution by strategically increasing the positive examples and/or removing the negative examples [8,17]. Cost-sensitive learning penalizes the misclassification of the minority class more to bias the learning towards the minority class intentionally [13,20]. However, effective cost information is usually not available. Ensemble learning with resampling obtains diverse base learners by employing over-sampled or under-sampled subsets and integrates them by boosting or bagging [3,6].

These algorithmic-level methods either introduce misclassification costs or simply rebalance class distribution without considering intrinsic complicated distributions of the minority class. In this paper, we aim to increase the accuracy of the minority class by dealing with small disjuncts and borderline examples. We propose a general approach, optimization embedded bagging (OEBag), by employing a decision tree to learn the complex distributions of the minority class more precisely. In learning the base classifiers, OEBag optimizes decision trees iteratively by learning the minority class examples that are easily misclassified. In addition to the overall evaluation metric, we also utilize the minority class performance metric, i.e., sensitivity as the optimization metric. OEBag is implemented by employing two under-sampling bagging methods.

The proposed method has several advantages. The increased sensitivity is superior in imbalanced classification. This benefits from the fact that OEBag focuses more on minority class learning. Furthermore, OEBag performs well in solving small disjuncts and overlapping between classes, and it shows robustness against complex data distributions and diverse imbalance levels. Therefore, OEBag increases the sensitivity and has a great overall performance. In addition, the out-of-bag examples are not discarded. They act as the reference base for the decision tree to choose examples for optimization.

The rest of this paper is organized as follows. Section 2 describes the proposed OEBag using two specialized under-sampling bagging methods. In Section 3, experimental setups and experimental results on 19 real datasets are reported. Section 4 gives the conclusions.

2. Optimization Embedded Bagging

2.1. Motivations

In this paper, the decision tree is employed as the base learner, which divides data space into homogeneous areas and builds the rules covering these areas. The difficult factors, small disjuncts and borderline examples mainly discussed in this paper can hinder the learning of the minority class and thus, degrade the classification performance of the minority class. Specifically, the decision tree tends to scatter the examples in small disjuncts and form data fragments, and it will increase the difficulty in learning the minority class. Furthermore, sampling can exclude some examples in small disjuncts when building each decision tree in bagging. In this paper, we use the impurity reduction as the split criterion and the Gini index as the measure of the node impurity; i.e., the Gini index minimization is enforced to choose the optimal feature and the optimal cut point. Hence, for borderline examples, especially in overlapping areas, the decision tree tends to bias towards the predominant class and leads the positive examples to reach the leaf node that is labelled as the negative class.

Therefore, we try to find the easily misclassified minority examples and learn them intentionally. This is achieved by using out-of-bag examples as a reference base and using sensitivity as the evaluation metric. In the framework of traditional bagging, several base learners are built and each base learner is trained using an under-sampled subset of the original training examples. Hence, for these base learners, different subsets of the original training examples are excluded. When we optimize a base learner, we use the excluded examples corresponding to the current base learner as a reference base. We call the excluded examples as out-of-bag examples.

2.2. Methods

Algorithm 1 is the pseudocode of OEBag. Given a training set Tr with l examples, each example consists of m attributes. Tr is composed of the positive (minority class) subset Pos and negative (majority class) subset Neg; i.e., $Tr = Pos \cup Neg$. In each of T iterations of OEBag, the maximum G-mean and sensitivity are initialized as 0. G-mean is the geometric mean of sensitivity and specificity [12]. Next, an initial subset is sampled from all training examples to learn a decision tree, and then all examples in the out-of-bag ($Tr \backslash Sub_t$ in line 5 of Algorithm 1) are classified. Specifically, the decision tree classifier is constructed based on the Gini index g (Equations (1)-(2)) of the attribute are as follows:

$$g(D_h, A) = \frac{|D_{hl}|}{|D_h|} g(D_{hl}) + \frac{|D_{hr}|}{|D_h|} g(D_{hr}) \tag{1}$$

$$g(D) = p^+(1 - p^+) + p^-(1 - p^-) \tag{2}$$

where $g(D_h, A)$ denotes the Gini index for the hth node with size $|D_h|$ for the attribute A, D_{hl} and D_{hr} denote the subsets of D_h corresponding to the left and right child nodes of the hth node split according to the attribute-value A = a, respectively; $g(D)$ indicates the uncertainty of set D, p^+ and p^- denote the proportion of positive and negative examples in set D, respectively. The attribute corresponding to the smallest Gini index will be selected to split the hth node. The process continues until the minimum example number of the leaf node is reached.

Algorithm 1 Optimization Embedded Bagging (OEBag)
Input: Tr, training set; Φ, the sampling method; DT, decision tree learning algorithm; EVA, evaluation method; T, number of iterations.
Output: H, OEBag ensemble classifier.
1: **for** t = 1 to T **do**
2: Initialization: $G\_max \leftarrow 0,\ sen\_max \leftarrow 0$;
3: $Sub_t \leftarrow \Phi(Tr)$;
4: $H_t \leftarrow DT(Sub_t)$;
5: $\{G,\ sen\} \leftarrow EVA(H_t,\ Tr \backslash Sub_t)$;
6: **while** $G \geq G\_max$ and $sen \geq sen\_max$ **do**
7: $G\_max \leftarrow G,\ sen\_max \leftarrow sen$;
8: Choosing misclassified minority examples in $Tr \backslash Sub_t$ and feeding them into Sub_t;
9: $H_t \leftarrow DT(Sub_t)$;
10: $\{G,\ sen\} \leftarrow EVA(H_t,\ Tr \backslash Sub_t)$;
11: **end while**
12: **end for**
13: $H \leftarrow majority\ vote\ of\ H_t$, where $t = 1,2,\ldots,T$.

Next, we select a misclassified minority example from each of the nodes where misclassified minority examples in the out-of-bag reach, and we put them in the current training subset for the next learning (line 8 of Algorithm 1) until G-mean and sensitivity do not increase any more. Since accuracy is not a good performance metric for imbalanced classification, we choose G-mean, a commonly used metric for imbalanced problems, to measure overall performance [12]. At last, all the optimized base classifiers are combined for the final decision with majority voting.

In this paper, the optimization is applied to RBBag and uNBBag, and we call the optimized methods OE-RBBag and OE-uNBBag, respectively. Since the sampling strategies of RBBag and uNBBag are different, the details specialized for OE-RBBag and OE-uNBBag will be explained. In [18], RBBag was performed using sampling with or without replacement, and experimental results have shown that their performances are similar. In OE-RBBag, we use sampling with replacement since we will use the minority examples in the out-of-bag to learn the positive class. The sampling in line 3 of Algorithm 1 is substituted as $Sub_t \leftarrow \Phi_{nbin}(Tr)$. Specifically, Φ_{nbin} samples subsets Pos_t and Neg_t from Pos and Neg randomly with replacement. Subsequently, we obtain the subset $Sub_t = Pos_t \cup Neg_t$. The size of each positive subset $|Pos_t|$ equals the number of the all positive examples in the training set $|Pos|$; i.e., $|Pos_t| = |Pos|$. The size of each negative subset $|Neg_t|$ is generated by the function $nbinrnd$, which obeys the negative binomial distribution. The parameters in the distribution are set as follows: the number of successes is $|Pos|$, and the probability of success is 0.5; i.e., $|Neg_t| = nbinrnd(|Pos|, 0.5)$.

The uNBBag method draws the sample subset in each iteration according to the weights considering the sample's importance. The principle is to assign larger weights to the minority examples than to the majority examples, and to assign

larger weights to less safe examples. Hence, the unsafe minority class examples tend to have larger weights and have higher probabilities to be drawn. Examples in bag, i.e., in weighted sampling subset, may construct a good decision tree for easy imbalanced datasets. However, for more difficult datasets, we still need to optimize the decision tree.

OE-uNBBag follows the weighted sampling proposed in [3], and the weights are defined according to the form shown in Equation (3)

$$w_{class} = g_{class} \times \beta(l_{class} + 1) \tag{3}$$

where $class$ represents the positive class pos or the negative class neg. The intuition behind the weight setting is that the more unsafe the example is, the more likely it should be drawn. Weights are at both global and local levels: at the global level, g_{class} is set as 1 or $|Pos|/|Neg|$ when the $class$ is positive or negative, respectively; at the local level, l_{neg} is set as 0, indicating that negative examples are more likely to be safe, whereas l_{pos} is set by counting the number of negative examples in the neighborhood. $l_{pos} = (k_{neg})^{\varphi}/k$, where k_{neg} is the number of negative example in k nearest neighbors ($k = 5$), and φ is a scaling factor ($\varphi = 0.5$). Notice that $k = 5$ is adopted from [3], and $\varphi = 0.5$ is chosen by empirical results. β is set to 0.5 to keep an equal probability of being drawn when an example from the minority class or majority class is safe. Therefore, the weights for positive and negative examples are shown in Equations (4) and (5), respectively:

$$w_{pos} = 0.5(\frac{(k_{neg})^{\varphi}}{k} + 1) \tag{4}$$

$$w_{neg} = 0.5 \times \frac{|Pos|}{|Neg|} \tag{5}$$

For OE-uNBBag, the weight w of training examples should be obtained before ensemble learning. The sampling in line 3 of Algorithm 1 is replaced as $Sub_t \leftarrow \Phi_{weighted}(Tr)$. Specifically, $\Phi_{weighted}$ draws $2|Pos|$ examples from the training set Tr according to weight w.

The learning of individual classifiers in OEBag can be considered as an evaluation metric driven process. Algorithmic development should be application-dependent. For imbalanced classification, since we want to put more focus on the minority class, we develop the optimization process in OEBag by evaluating the sensitivity. In addition, to avoid poor performance of the majority class, we also evaluate G-mean, a popularly used comprehensive metric in imbalanced classification [3,15,20].

3. Experimental framework

3.1. Experimental datasets and setups

We use 19 real datasets from the UCI repository [1]. For datasets with more than two classes, we choose the small one as the minority class and combine the rest as the majority class, as done in [3]. The characteristics of the 19 datasets are listed in Table 1. Ins, Min, IR, and Attr denote sample size, minority class size, imbalance ratio and attribute dimension, respectively. These datasets are categorized into several groups according to the types of examples that form the minority class distribution [14]. The type of each dataset is defined as the prevailing type of the minority class examples: safe (S), borderline (B), rare (R) and outlier (O). Safe examples are located in relatively homogeneous areas, whereas borderline examples occur in the area surrounding the class boundary, no matter whether there is overlapping or no overlapping between the minority and majority classes. Rare cases are the few isolated examples in the opposite class space, which are distant from the class border. The outlier is a single example, representing a rare but valuable sub-concept.

Table 1. Characteristics of the real datasets

Type	Data Set	Ins	Min	IR	Attr	Type	Data Set	Ins	Min	IR	Attr
S	breast-w	699	239	1.90	9	R,O	cmc	1473	333	3.42	9
	new-thyroid	215	35	5.14	5		cleveland	303	35	7.66	13
	vehicle	846	199	3.25	18		abalone	4177	335	11.47	8
S,B	car	1728	69	24.04	6		postoperative	90	24	2.75	8
	ionosphere	351	126	1.79	34	O	solar-flare	1066	43	23.79	12
	pima	768	268	1.87	8		transfusion	748	178	3.2	4
B,R	credit-g	1000	300	2.33	20		yeast	1484	51	28.1	8
	ecoli	336	35	8.6	7		balance-scale	625	49	11.76	4
	hepatitis	155	32	3.84	19						
	haberman	306	81	2.78	3						
	breast-cancer	286	85	2.36	9						

Table 2. Results of AUC, sensitivity, and G-mean of RBBag and OE-RBBag

	AUC		sensitivity		G-mean	
Dataset	RBBag	OE-RBBag	RBBag	OE-RBBag	RBBag	OE-RBBag
breast-w	**0.9868**	0.9861	0.9615	**0.9782**	0.9578	**0.9638**
new-thyroid	**0.9884**	0.9876	0.9486	**0.9600**	**0.9335**	0.9240
vehicle	**0.9859**	0.9846	0.9789	**0.9859**	**0.9454**	0.9417
car	0.9615	**0.9664**	1.0000	**1.0000**	**0.8879**	0.8841
ionosphere	**0.9456**	0.9406	0.8794	**0.8810**	**0.8918**	0.8917
pima	**0.8299**	0.8249	0.7679	**0.8075**	0.7587	**0.7606**
credit-g	0.7758	**0.7776**	0.7160	**0.7600**	**0.7069**	0.7055
ecoli	0.9257	**0.9288**	0.9486	**0.9543**	**0.8586**	0.8576
hepatitis	0.8823	**0.8854**	**0.9077**	0.8769	**0.7763**	0.7563
haberman	**0.7066**	0.6963	0.6173	**0.6691**	**0.6505**	0.6451
breast-cancer	**0.6797**	0.6693	0.6272	**0.6519**	**0.6472**	0.6192
cmc	0.7327	**0.7360**	0.6571	**0.7141**	0.6708	**0.6708**
cleveland	0.8096	**0.8194**	0.7829	**0.8114**	**0.7585**	0.7415
abalone	**0.8791**	0.8783	0.7713	**0.8012**	**0.7973**	0.7960
postoperative	**0.3923**	0.3846	0.4750	**0.5083**	**0.4299**	0.3790
solar-flare	0.8807	**0.8873**	0.8744	**0.8837**	0.8294	**0.8336**
transfusion	**0.7365**	0.7271	0.6427	**0.6753**	**0.6826**	0.6763
yeast	0.9085	**0.9132**	0.8627	**0.9373**	**0.8167**	0.8161
balance-scale	0.4594	**0.5047**	0.3673	**0.5592**	0.4347	**0.4763**
Average	0.8141	**0.8157**	0.7782	**0.8113**	**0.7597**	0.7547

Here, we use AUC [4], sensitivity and G-mean as performance evaluation metrics. AUC is the area under the Receiver Operating Characteristic (ROC) curve, which is depicted using the false positive rate as the x-axis and the true positive rate as the y-axis. AUC has been recommended in many works about imbalanced learning [2,17]. However, we cannot find a method that can obtain the highest AUC or G-mean values for all of the datasets. We claim that when one method outperforms other competitors in most cases on comprehensive metrics, obtaining high sensitivity (true positive rate) should be the main objective.

The experimental results are obtained by performing a stratified 10-fold cross validation. For each time, 9 folds (training set) are used to train ensemble classifiers, and the remaining fold (test set) is used to evaluate the performance of the classifier; the procedure is repeated 10 times until each fold has been used as the test set. The whole cross validation process is repeated 5 times, and the final values of AUC, sensitivity, and G-mean are the averaged results of the 5 times. The number of iterations T is set as 30. The decision tree is built without pruning and the node is split according to the Gini index. The minimum number of examples per leaf in the decision tree is set as 10.

3.2. Comparisons before and after optimization

Tables 2 and 3 show the results of RBBag vs. OE-RBBag and uNBBag vs. OE-uNBBag on AUC, sensitivity, and G-mean. The better value for each dataset is highlighted in bold. From the two tables, we can observe that after optimization, OE-RBBag and OE-uNBBag win on most of the datasets in terms of sensitivity. Wilcoxon signed rank tests for RBBag vs. OE-RBBag and uNBBag vs. OE-uNBBag are also performed. The p-values of the comparisons associated with sensitivity are smaller than 0.05, indicating significant improvements on sensitivity after optimization. The comparisons on AUC and G-mean do not show significant differences. In such a case, OE-RBBag (OE-uNBBag) obtains a comparative overall performance and much better sensitivity than RBBag (uNBBag).

3.3. Comparisons of OEBag methods and boosting methods

Here, we compare the performance of OE-RBBag and OE-uNBBag vs. under-sampling boosting method RUSBoost and cost-sensitive boosting method AdaC3. Since [6] has shown that under-sampling boosting performs better than over-sampling boosting, we choose RUSBoost as the representative of under-sampling boosting ensemble methods. For cost-sensitive methods, we choose AdaC3 in the three AdaC methods [20]. The learning procedures of RUSBoost and AdaC3 are the same

as that of OEBag methods. The parameters of the number of iterations and the minimum number of examples per leaf in the decision tree are the same as described in Section 3.1. Other experimental setups about RUSBoost and AdaC3 are as follows: in RUSBoost, RUS removes the majority class examples until the numbers of both classes are equal; referring to the original work of AdaC3 [20], we set the costs of the positive class and the negative class as 1 and 0.6, respectively.

Table 3. Results of AUC, sensitivity, and G-mean of uNBBag and OE-uNBBag

	AUC		sensitivity		G-mean	
Dataset	uNBBag	OE-uNBBag	uNBBag	OE-uNBBag	uNBBag	OE-uNBBag
breast-w	0.9845	**0.9867**	0.9708	**0.9832**	0.9531	**0.9677**
new-thyroid	**0.9958**	0.9884	**0.9417**	0.9083	**0.9377**	0.8885
vehicle	0.9811	**0.9842**	0.9845	**1.0000**	0.9391	**0.9392**
car	**0.9639**	0.9581	**1.0000**	**1.0000**	**0.8861**	0.8838
ionosphere	0.9422	**0.9565**	0.8788	**0.8962**	0.8952	**0.9000**
pima	0.8187	**0.8220**	0.8164	**0.8241**	**0.7413**	0.7354
credit-g	0.7734	**0.7738**	0.7667	**0.8000**	**0.6850**	0.6845
ecoli	**0.9393**	0.9111	**0.9500**	0.9333	**0.8505**	0.8413
hepatitis	0.8643	**0.8792**	**0.8500**	**0.8500**	0.6841	**0.7837**
haberman	0.7063	**0.7079**	0.7167	**0.7417**	**0.6441**	0.6249
breast-cancer	0.6669	**0.6719**	0.6889	**0.7153**	**0.5770**	0.5512
cmc	0.7319	**0.7397**	0.7566	**0.7990**	0.6598	**0.6607**
cleveland	0.7993	**0.8214**	0.8167	**0.8667**	0.7213	**0.7278**
abalone	**0.8818**	0.8810	0.8118	**0.8295**	**0.7882**	0.7865
postoperative	0.3514	**0.4637**	**0.6167**	**0.6167**	**0.2790**	0.2041
solar-flare	**0.8980**	0.8821	0.8850	**0.8950**	0.8275	**0.8348**
transfusion	0.7214	**0.7421**	0.7137	**0.7752**	0.6354	**0.6554**
yeast	0.9057	**0.9126**	0.9033	**0.9633**	0.7971	**0.8063**
balance-scale	**0.5082**	0.4738	0.7800	**0.7950**	**0.3734**	0.3347
Average	0.8123	**0.8187**	0.8341	**0.8522**	**0.7303**	0.7269

Table 4 shows the results of two OEBag methods and two boosting methods in terms of AUC, sensitivity, and G-mean. The best value for each dataset is highlighted in bold. Note that from top to bottom, the real datasets are listed from easy to hard. We can observe that AdaC3 performs best on easy datasets considering the three metrics. However, for hard datasets such as *yeast*, the sensitivity is very small, whereas for the *transfusion* and *postoperative* datasets, the sensitivity is very large and the G-mean is very small. This may attribute to inappropriate cost terms. The drawbacks of cost-sensitive methods are that cost terms are difficult to obtain and the appropriate costs are different for datasets. The OEBag methods perform best on hard datasets, which benefits from the optimization procedure by considering the complicated distributions when learning base classifiers. OE-uNBBag obtains better AUC and sensitivity than OE-RBBag. This is because OE-uNBBag employs the weighted sampling, which focuses more on the minority class than on the majority class. As a result, the specificity values of OE-uNBBag decrease a little, and the G-mean values of OE-uNBBag are a bit lower than that of OE-RBBag on average. Note that G-mean is the geometric mean of sensitivity and specificity.

For comparing the four methods clearly, we also use the Aligned Friedman statistical test [7] to compute the average rankings and the p-values associated with the significance of differences among the methods. If the p-values are smaller than 0.05, there are significant differences among the compared methods. Then, a post-hoc test is performed to identify which method performs best significantly and the adjusted p-values of Hochberg's test [9] are computed. The statistical test results are shown in Table 5. We can observe that the statistical test results are in accordance with the results shown in Table 4. OE-RBBag and OE-uNBBag do not show significant differences on the three metrics. Compared with OEBag methods, AdaC3 shows no significant difference on sensitivity, but exhibits significantly poor results on AUC and G-mean. OEBag methods perform significantly better than RUSBoost on AUC and sensitivity.

Table 4. Comparisons of AUC, Sensitivity, and G-mean between OEBag methods and two boosting methods

	AUC				sensitivity				G-mean			
Dataset	RUSBoost	AdaC3	OE-RBBag	OE-uNBBag	RUSBoost	AdaC3	OE-RBBag	OE-uNBBag	RUSBoost	AdaC3	OE-RBBag	OE-uNBBag
breast-w	0.9786	**0.9918**	0.9861	0.9867	0.9397	0.9724	0.9782	**0.9832**	0.9428	0.9675	0.9638	**0.9677**
new-thyroid	0.9917	**0.9974**	0.9876	0.9884	0.9371	0.9257	**0.9600**	0.9083	0.9419	**0.9578**	0.9240	0.8885
vehicle	0.9767	**0.9966**	0.9846	0.9842	0.9387	0.9749	0.9859	**1.0000**	0.9359	**0.9727**	0.9417	0.9392
car	0.9386	**0.9993**	0.9664	0.9581	0.8261	0.9536	**1.0000**	**1.0000**	0.8635	**0.9744**	0.8841	0.8838
ionosphere	0.9308	**0.9694**	0.9406	0.9565	0.8175	**0.9048**	0.8810	0.8962	0.8580	**0.9183**	0.8917	0.9000
pima	0.7941	0.7690	**0.8249**	0.8220	0.7187	**0.9045**	0.8075	0.8241	0.7288	0.6314	**0.7606**	0.7354
credit-g	0.7535	0.7382	**0.7776**	0.7738	0.6840	**0.8760**	0.7600	0.8000	0.6899	0.5990	**0.7055**	0.6845
ecoli	0.9094	**0.9362**	0.9288	0.9111	0.7486	0.7486	**0.9543**	0.9333	0.8393	0.8357	**0.8576**	0.8413
hepatitis	0.9023	0.8572	**0.8854**	0.8792	0.7692	0.8154	**0.8769**	0.8500	**0.8015**	0.7757	0.7563	0.7837
haberman	0.6693	0.6385	0.6963	**0.7079**	0.5136	**0.9951**	0.6691	0.7417	0.6160	0.2049	**0.6451**	0.6249
breast-cancer	0.6364	0.6331	0.6693	**0.6719**	0.5012	**0.9630**	0.6519	0.7153	0.5969	0.2628	**0.6192**	0.5512
cmc	0.7071	0.6682	0.7360	**0.7397**	0.6324	**0.9405**	0.7141	0.7990	0.6562	0.3814	**0.6708**	0.6607
cleveland	0.7220	0.7210	0.8194	**0.8214**	0.4286	0.5143	0.8114	**0.8667**	0.6130	0.6302	**0.7415**	0.7278
abalone	0.8261	0.8420	0.8783	**0.8810**	0.7475	0.4812	0.8012	**0.8295**	0.7737	0.6710	**0.7960**	0.7865
postoperative	0.4095	0.3958	0.3846	**0.4637**	0.1167	**1.0000**	0.5083	0.6167	0.3046	0.0000	**0.3790**	0.2041
solar-flare	0.8508	0.8551	**0.8873**	0.8821	0.7349	0.8372	0.8837	**0.8950**	0.7990	0.7900	0.8336	**0.8348**
transfusion	0.6974	0.6632	0.7271	**0.7421**	0.5663	**0.9854**	0.6753	0.7752	0.6603	0.1621	**0.6763**	0.6554
yeast	0.7883	0.9036	**0.9132**	0.9126	0.7216	0.2980	0.9373	**0.9633**	0.8015	0.5411	**0.8161**	0.8063
balance-scale	0.5282	**0.6804**	0.5047	0.4738	0.1224	**0.9102**	0.5592	0.7950	0.3295	**0.6400**	0.4763	0.3347
Average	0.7856	0.8021	**0.8202**	0.8187	0.7146	0.7511	0.8139	**0.8522**	0.7215	0.6640	**0.7335**	0.7269

Table 5. Rankings of the aligned Friedman test and the adjusted P-values with Hochberg test

	AUC		sensitivity		G-mean	
Method	Ranking	$\mathbf{P_{Hochberg}}$	Ranking	$\mathbf{P_{Hochberg}}$	Ranking	$\mathbf{P_{Hochberg}}$
RUSBoost	52.7368	6.79E-04	60.3421	2.48E-06	37.5789	0.2458
AdaC3	45.4737	0.0150	32.5526	0.2935	52.8421	0.0013
OE-RBBag	29.4737	0.6594	36.0789	0.2458	27.6316	--
OE-uNBBag	26.3158	--	25.0263	--	35.9474	0.2458
$\mathbf{P_{AlignedFriedman}}$	**0.001639**		**0.001623**		**0.0020**	

4. Conclusions

We propose an optimized bagging approach OEBag with a decision tree as the base learner to bias the imbalanced learning towards the minority class intentionally, which is realized by intensively dealing with small disjuncts and borderline examples in the minority class. Comparisons with the state-of-the-art ensemble methods on real datasets demonstrate OEBag's superiority in improving the sensitivity; meanwhile, OEBag ensures a comparative overall performance in terms of AUC and G-mean.

The fact that OEBag achieves better results than those of the competitors validates an important conclusion: taking complex distributions into consideration is beneficial for imbalanced classification. Hence, in future studies we should explore intrinsic distributions in imbalanced datasets and develop a general learning strategy to improve the performance, instead of using a decision tree alone. In the future, we expect to expand the proposed method into the field of big data and practical applications, since one of the main features of big data is unbalance.

References

1. A. Asuncion, and D. Newman, "UCI Machine Learning Repository," 2007
2. G. E. Batista, R. C. Prati, and M. C. Monard, "A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data," *ACM Sigkdd Explorations Newsletter,* vol. 6, no. 1, pp. 20-29, 2004
3. J. Błaszczyński, and J. Stefanowski, "Neighbourhood Sampling in Bagging for Imbalanced Data," *Neurocomputing,* vol. 150, pp. 529-542, 2015
4. A. P. Bradley, "The Use of the Area under the ROC Curve in the Evaluation of Machine Learning Algorithms," *Pattern Recognition,* vol. 30, no. 7, pp. 1145-1159, 1997
5. M. Galar, A. Fernández, E. Barrenechea, H. Bustince, and F. Herrera, "Ordering-Based Pruning for Improving the Performance of Ensembles of Classifiers in the Framework of Imbalanced Datasets," *Information Sciences,* vol. 354, pp. 178-196, 2016
6. M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera, "A Review on Ensembles for the Class Imbalance Problem: Bagging-, Boosting-, and Hybrid-Based Approaches," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on,* vol. 42, no. 4, pp. 463-484, 2012
7. S. García, A. Fernández, J. Luengo, and F. Herrera, "Advanced Nonparametric Tests for Multiple Comparisons in the Design of Experiments in Computational Intelligence and Data Mining: Experimental Analysis of Power," *Information Sciences,* vol. 180, no. 10, pp. 2044-2064, 2010
8. H. Guan, Y. Zhang, M. Xian, H. Cheng, and X. Tang, "WENN for Individualized Cleaning in Imbalanced Data," In *Pattern Recognition (ICPR), 2016 23rd International Conference on*, pp. 456-461, IEEE
9. Y. Hochberg, "A Sharper Bonferroni Procedure for Multiple Tests of Significance," *Biometrika,* vol. 75, no. 4, pp. 800-802, 1988
10. R. C. Holte, L. Acker, and B. W. Porter, "Concept Learning and the Problem of Small Disjuncts," In *IJCAI*, pp. 813-818, Citeseer
11. N. Japkowicz, and S. Stephen, "The Class Imbalance Problem: A Systematic Study," *Intell Data Anal,* vol. 6, no. 5, pp. 429-449, 2002
12. V. López, A. Fernández, S. García, V. Palade, and F. Herrera, "An Insight into Classification with Imbalanced Data: Empirical Results and Current Trends on Using Data Intrinsic Characteristics," *Information Sciences,* vol. 250, pp. 113-141, 2013
13. C. X. Ling, Q. Yang, J. Wang, and S. Zhang, "Decision Trees with Minimal Costs," In *Proceedings of the Twenty-First International Conference on Machine Learning*, pp. 69, ACM
14. K. Napierala, and J. Stefanowski, "Types of Minority Class Examples and Their Influence on Learning Classifiers from Imbalanced Data," *Journal of Intelligent Information Systems,* pp. 1-35, 2015
15. K. Napierała, J. Stefanowski, and S. Wilk, "Learning from Imbalanced Data in Presence of Noisy and Borderline Examples," In *Rough Sets and Current Trends in Computing*, pp. 158-167, Springer
16. R. C. Prati, G. E. Batista, and M. C. Monard, "Class Imbalances Versus Class Overlapping: An Analysis of a Learning System Behavior," MICAI 2004: Advances in Artificial Intelligence, pp. 312-321, Springer, 2004
17. J. A. Sáez, J. Luengo, J. Stefanowski, and F. Herrera, "SMOTE–IPF: Addressing the Noisy and Borderline Examples Problem in Imbalanced Classification by a Re-Sampling Method with Filtering," *Information Sciences,* vol. 291, no. pp. 184-203, 2015
18. H. Shohei, K. Hisashi, and T. Yutaka, "Roughly Balanced Bagging for Imbalanced Data," *Statistical Analysis & Data Mining,* vol. 2, no. 2, pp. 412-426, 2009
19. J. Stefanowski, "Overlapping, Rare Examples and Class Decomposition in Learning Classifiers from Imbalanced Data," Emerging Paradigms in Machine Learning, pp. 277-306, Springer, 2013
20. Y. Sun, M. S. Kamel, A. K. C. Wong, and Y. Wang, "Cost-Sensitive Boosting for Classification of Imbalanced Data," *Pattern Recognition,* vol. 40, no. 12, pp. 3358-3378, 2007
21. A. Tesfahun and D. L. Bhaskari, "Intrusion Detection Using Random Forests Classifier with SMOTE and Feature Reduction," In *International Conference on Cloud & Ubiquitous Computing & Emerging Technologies*, pp. 127-132
22. H. L. Yu, and J. Ni, "An Improved Ensemble Learning Method for Classifying High-Dimensional and Imbalanced Biomedicine Data," *IEEE-ACM Transactions on Computational Biology and Bioinformatics,* vol. 11, no. 4, pp. 657-666, 2014

Hongjiao Guan is a Ph.D. student of Pattern Recognition and Intelligence System from Harbin Institute of Technology, Harbin, China. Her research interests include pattern recognition and machine learning.

Yingtao Zhang received her M.S. degree from the Computer Science School of Harbin Institute of Technology, Harbin, China, in 2004, and her Ph.D. degree in Pattern Recognition and Intelligence System from Harbin Institute of Technology,

Harbin, China, in 2010. Now, she is an Associate Professor at the School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China. Her research interests include pattern recognition, computer vision and medical image processing.

Hengda Cheng received his Ph.D. degree in Electrical Engineering from Purdue University, West Lafayette, IN, in 1985, under the supervision Prof. K. S. Fu. He is currently a Full Professor with the Department of Computer Science, and an Adjunct Full Professor with the Department of Electrical Engineering, Utah State University, Logan, UT. He is an Adjunct Professor and a Doctorial Supervisor with the Harbin Institute of Technology. He is also a Guest Professor with the Institute of Remote Sensing Application, Chinese Academy of Sciences, Wuhan University, as well as Shantou University, and a Visiting Professor with Northern Jiaotong University.

Xianglong Tang is a Professor and Ph.D. supervisor at the school of Computer Science and Technology, Harbin Institute of Technology. His research interests include artificial intelligence and information processing.