A Random Forest with Minority Condensation and Decision Trees for Class Imbalanced Problems

SUVAPORN HOMJANDEE, KRUNG SINAPIROMSARAN
Department of Mathematics and Computer Science, Faculty of Science
Chulalongkorn University
Phayathai, Phatumwan, Bangkok
THAILAND

Abstract: Building an effective classifier that could classify a target or class of instances in a dataset from historical data has played an important role in machine learning for a decade. The standard classification algorithm has difficulty generating an appropriate classifier when faced with an imbalanced dataset. In 2019, the efficient splitting measure, minority condensation entropy (MCE) [1] is proposed that could build a decision tree to classify minority instances. The aim of this research is to extend the concept of a random forest to use both decision trees and minority condensation trees. The algorithm will build a minority condensation tree from a bootstrapped dataset maintaining all minorities while it will build a decision tree from a bootstrapped dataset of a balanced dataset. The experimental results on synthetic datasets apparent the results that confirm this proposed algorithm compared with the standard random forest are suitable for dealing with the binary-class imbalanced problem. Furthermore, the experiment on real-world datasets from the UCI repository shows that this proposed algorithm constructs a random forest that outperforms other existing random forest algorithms based on the recall, the precision, the F-measure, and the Geometric mean.

Key-Words: Class imbalanced, classification, minority condensation entropy, random forest

Received: February 27, 2021. Revised: August 23, 2021. Accepted: September 10, 2021. Published: September 19, 2021.

1 Introduction

Classification is one of the machines learning supervised processes to predict the class instances from future data with a given historical dataset [2]. However, classification algorithms have difficulty generating an appropriate classifier when faced with real-world datasets having a small number of instances for some classes. This generally is called a class imbalanced problem. A class imbalanced problem deals with identifying a small proportion of instances in a class correctly among other instances of other classes in the same dataset. In binary classification, the majority class is normally represented by the negative class, while the minority class is represented by the positive class. In the real-world problems, the minority class is frequently more important and receives much attention to correctly classify. For example, in credit card fraud detection, there is a small number of fraudulent transactions, but they are unusual and must be discovered. In the same way as disease diagnosis [3], the prediction of disease patients is more significant than normal people.

Many methods have been presented to deal with the class imbalanced problem where it can be categorized into four different approaches [4] there are 1) a data-level approach 2) an algorithmic-level approach 3) a cost-sensitive approach, and 4) an ensemble approach. This research interest is the study of an algorithm-level approach since it does not cause any shift in data distribution, and it is more adaptable to various characteristics of imbalanced datasets. In addition, the idea of developing the algorithm to build the random forest classifier that is suitable for classifying an imbalanced dataset is one of the methods that have received wide attention.

The random forest algorithm is a collection of small decision trees with sampling features to classify instances and has two important steps that are (1) a bootstrap on a training set and (2) building different decision trees from subsamples and randomized attributes. Nevertheless, when the bootstrap is used on an imbalanced dataset, there is a chance that most minorities will not be picked during the bootstrapping step. Thus, the development of a decision tree algorithm that can handle this problem from the bootstrapping step, will make the prediction of the random forest more efficient. Recently, MCE is proposed as the splitting measure in partitioning algorithms to build the decision tree for handling the binary-class imbalanced numeric dataset and the decision tree that is built based on MCE is called the minority condensation decision tree or MCDT. Hence, these

are motivated for enhancement of the random forest algorithm of this research that is extending MCDT to construct the random forest.

2 Background

2.1 Minority Condensation Decision Tree

MCE is entropy that is modified from minority entropy (ME) [5], used to find the best attribute for constructing a decision tree. MCE is designed to handle a binary-class imbalanced dataset. Initially, in this research, we give a binary-class dataset that consists of instances from a positive class and negative class, i.e., given the dataset D, the attribute $a \in \{A_1, A_2, \ldots, A_M\}$ represent the selected attribute. D consists of instances from a set of two classes $C = \{+, -\}$. In the standard decision tree, Shannon's entropy (SE) [6], the splitting measure, is computed based on the impurity of each partition, denote by $Entropy$, the formulation in Equation 1.

$$Entropy(D) = -\frac{|D_+|}{|D|} log_2 \frac{|D_+|}{|D|} - \frac{|D_-|}{|D|} log_2 \frac{|D_-|}{|D|} \quad (1)$$

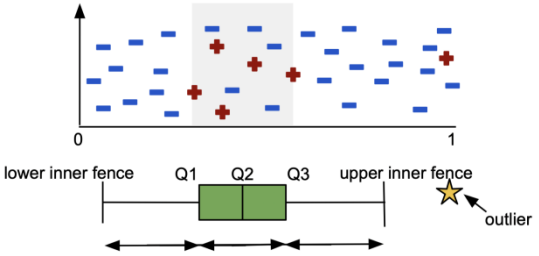


Fig. 1: Applying the IQR rule to detect outliers before determining the minority instances

The computation of MCE is based on the interquartile range (IQR) rule that is employed to the set of minority instance values for detecting the outlines. It defines the boundary that represents the range of acceptable values for the minority instances based on Tukey's boxplot [7]. The lower inner fence is defined by the first quartile minus 1.5 times of IQR, while the upper inner fence is defined by the third quartile plus 1.5 times of IQR. For example, Figure 1 demonstrates the use of the IQR rule. The set of instances within that range is considered, in which the minority class is more condensed. Accordingly, SE from Eq.1 computed with that set is called MCE, and then the decision tree is built based on MC is called minority condensation decision tree (MCDT).

2.2 Random Forest

Random forest [8] is classified as a supervised machine learning algorithm that has come into the limelight recently. The decision tree forms the base classifier in a random forest. This classifier combines the predictions made by multiple decision trees. As the named randomization is done in two ways in constructing random forests. One is using random sampling or bootstrapping for drawing subsamples and the second is randomly selecting attributes or features for generating decision trees.

The steps in constructing the decision tree in the forest are 1) Take D as the number of training data instances in the samples, then let M be the number of attributes in the input dataset, and m represent the number of attributes to choose at each tree node. 2) The training samples are gathered, and for each subsample, a replacement tree is constructed. 3) For each tree node, choose m attributes at random. 4) The optimal split is computed based on the m input attributes of the subsampled dataset, and 5) Each tree is allowed to grow without being pruned. Then, the most predictions from these trees will then be used to determine the final.

3 Motivation and Methodology

The motivation of this research comes from the success of using the decision tree classifier based on MCE to handle the class imbalanced problem. It fixed the problem of the ME that sometimes it unnecessarily widens the minority range, which covers more majority instances because of having minority instance values extremely deviate from others within datasets. These reasons make MCDT highly successful in handling the class imbalanced problem and from this performance can be improved with the ensemble learning method, in which multiple decision trees are combined as a random forest classifier. However, from bootstrapping of random forest, when the algorithm is faced with an imbalanced dataset, it has a chance to make minority instances disappear and the prediction performances of the classifier are decreasing. As the result, the idea of keeping all minorities for bootstrapping comes from these reasons. However, the standard decision tree still outperforms for prediction a balanced dataset. Therefore, this research proposes a random forest that uses a mixed decision tree and MCDT, and then we will call this enhanced random forest is RMDT, which has 2 construction parts that are

1. It will bootstrap only majority instances and keep all minority instances in a training dataset's bootstrapping phase, ignoring the balance of these two classes, therefore subsamples from this part are

imbalanced datasets, and MCE should be used to discover the best attribute to split in each decision tree.

2. Bootstrapping also maintains all minority instances and bootstraps just majority instances, however in this section, we bootstrap until the number of majority instances and minority instances are equal, resulting in a dataset with balanced subsamples. The decision tree will then be built using SE.

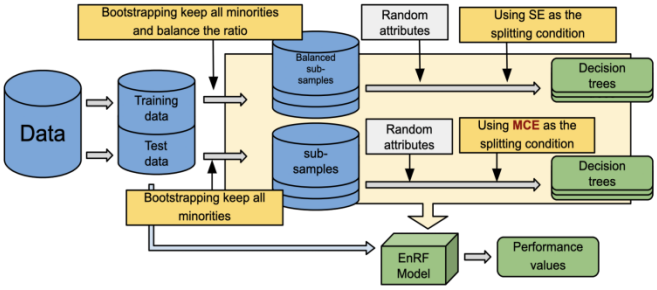


Fig. 2: A framework of the enhancing random forest algorithm

Visualization representation of these two parts is explained in the following Figure 2 and the algorithm construction is described in Algorithm 1. The following parameters are required by the algorithm: N (number of standard trees trained in the forest) and n (number of minority condensation decision trees trained in the forest). Subsample computing is performed in Line 6 of Algorithm 1 for use in Line 8 and Line 11 which are balanced subsamples, and imbalanced subsamples, respectively.

Algorithm 1: RMDT (D, N, n)

Input:
1: Learning data D
2: User specified values of the number of DT,
3: the number of MCDT
Learning Phase:
4: **procedure** RMDT(D, N, n)
5: Let $T = n + N$
6: **for** each t_i in T **do**
7: s = compute for balanced/ imbalanced subsamples
8: **if** $t_i \leq n$ then
9: F1 = built forest base on subsamples for
10: standard decision trees
11: **else**
12: F2 = built forest base on subsamples for
13: MCDT
14: **end for**
15: RMDT = F1∪F2{combined two forests to main
16: forest}
17: **end procedure**

4 Experiment and Results

4.1 Performance Measure and Evaluation

The efficiency of a classifier in a class imbalanced problem is evaluated quantitatively based on the precision and recall, which are derived from the confusion matrix (Table 1). In this table, true positive (TP) denotes the number of positive instances that are correctly predicted as positive instances, true negative (TN) denotes the number of negative instances that are correctly predicted as negative instances, false positive (FP) denotes the number of negative instances that are inaccurately predicted as positive instances and false negative (FN) denotes the number of positive instances that are inaccurately predicted as negative instances.

Table 1. Confusion Metric

	Actual positive	**Actual negative**
Predicted positive	True positive (TP)	False positive (FP)
Predicted negative	False negative (FN)	True negative (TN)

Recall exhibits how many relevant items are selected and precision exhibits how many selected items are relevant. According to [9], F-measure and Geometric mean are the performance measures that are suitable for a class imbalanced problem, which harmonizes recall in Eq.3 and precision in Eq.4. β is the weight of importance between recall and precision, it is set to 1 which means they are equally important. The formulae for the F-measure are provided in Eq.5.

$$Recall = \frac{TP}{TP+FN} \tag{2}$$

$$Precision = \frac{TP}{TP+FP} \tag{3}$$

$$F-measure = (1+\beta^2) \times \frac{Recall \times Precision}{\beta \times Recall + Precision} \tag{4}$$

$$Geometric\ mean = \sqrt{Recall \times Precision} \tag{5}$$

4.2 Datasets

The experiments were performed on 20 imbalanced binary datasets. Ten of these datasets are from synthetic binary-class imbalanced numeric datasets consisting of 500 instances having 50 attributes. Each class is formed as a gaussian cluster which is located around a centroid in two dimensions. For each cluster, informative attributes are drawn

independently from $N(0,1)$. The clusters are then placed on the centroids. There are ten groups of experiments having different percentages of minority instances from 5% to 50%, then repeating 20 times for each experiment.

Furthermore, ten real-world application datasets from the UCI repository [10] are used. In Table 2, they are sorted in descending order by the percentage of instances in the minority class (%Min.). The first two columns indicate the number and the name of each dataset. For the number of instances (#Inst.) and
the number of attributes (#Att.), they are shown in the

Table 2. The characteristics of real-world binary-class datasets used in the experiments

No	Datasets	#Inst	#Att	Min/Maj	%Min	I.R.
1	Pima	768	8	'1'/ '0'	34.9	1.87
2	StakotlogVeh-icle	846	18	'bus'/The rest	25.77	2.88
3	BeastTissue	106	9	'fad'/The rest	14.15	6.07
4	NewThyroid	215	5	'3'/The rest	13.95	6.17
5	Fertility	100	9	'O'/ 'N'	12	7.33
6	Ecoli	336	7	'imU'/The rest	10.42	8.6
7	OpticDigits	1108	641	'8'/ The rest	9.86	9.14
8	Glass	214	9	'5'/The rest	6.07	15.46
9	winequality-red	1599	11	'3'/The rest	3.94	24.38
10	Yeast	1484	8	'VAC'/The rest	2.02	48.47

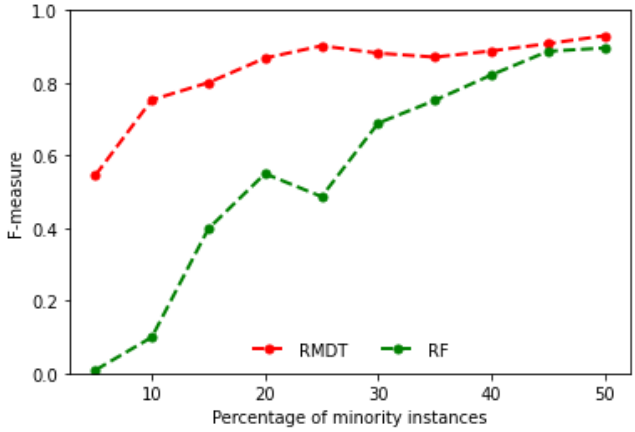
third column and the fourth column, respectively. Particularly, the minority class and the majority class are presented in the fifth column. In order to evaluate the performance of each method, the experiments are repeated 50 times. See Table 2 for their descriptions, including the number of instances, the number of features, the percentage of minorities, and the imbalanced ratio.

4.3 Experimental results and discussion

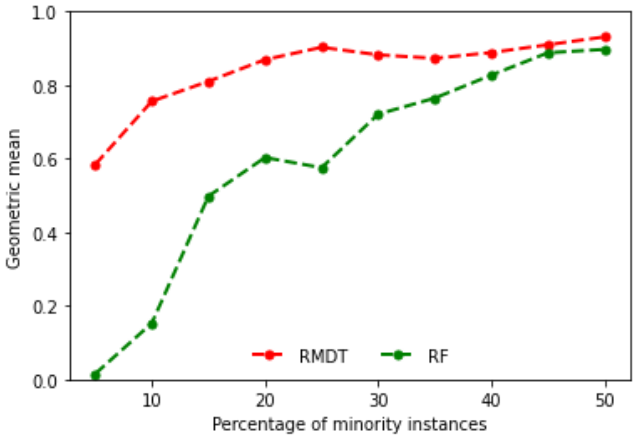
An enhancement of the standard random forest (RF) to classify minority instances in the binary-class imbalanced datasets dealing with numeric attributes using SE and MCE is exhibited in the experiments on collections of synthetics dataset according to section 4.2. Accordingly, the average results of RF and RMDT are compared via the F-measure (4) and the Geometric mean (5) with respect to the minority class and the majority class displaying in Figures 3(a) and 3(b) respectively.

For the results, the F-measure and the Geometric mean values of both RMDT (Red line) and RF (Green line) increase when the percentage of minority instances increases. Evidently, RMDT significantly outperforms RF when the number of minority instances is tiny, while their values will approach 1 when a dataset is more balanced. It is because RF tends to focus on the class having a large number of instances, while RMDT tries to make them balanced before considering. Therefore, these results confirm that RF is ineffective in dealing with binary-class imbalanced problems.

Moreover, to demonstrate the effectiveness of RMDT on a general dataset, the random forest built based on MCE and SE is evaluated with experiments on real-world datasets. The results are compared to those of three other classifiers. The first is RF. Additionally, the decision tree built based on MCE is used as well. Lastly, the popular boosting algorithm that works by weighting the instances, like AdaBoost [11, 12] is also used in the comparison. It increases the weight of instances that are difficult to classify and lowers the weight of instances that are easy to classify.



(a)



(b)

Fig.3: The experimental results on synthetic datasets varying percentage of minority instances comparing with Standard random forest (RF) via F-measure (a) and Geometric mean (b)

In order to evaluate the dataset into the training set and the testing set, they are repeated 10 times. Accordingly, the average results of each classifier are compared via the recall (2), the precision (3), the

F-measure (4), and the Geometric mean (5). Graphically, the bar chart representing the comparison of the average performance corresponding to each performance measure is shown in Figure 4, in which the higher value indicates the better performance, the green bar denotes RMDT performances values, orange denotes RF values, blue bar and pink bar denote MCDT and AdaBoost performance values, respectively. Comparing the precision of all classifiers, RMDT yields the highest average performance at 0.723, which is much different from AdaBoost, MCDT, and RF. They yield the fourth-highest, thirst-highest, and second-highest average performance at 0.473, 0.445, and 0.412, respectively. It means that the number of they predicted majority instances to be the minority class has more than RMDT.

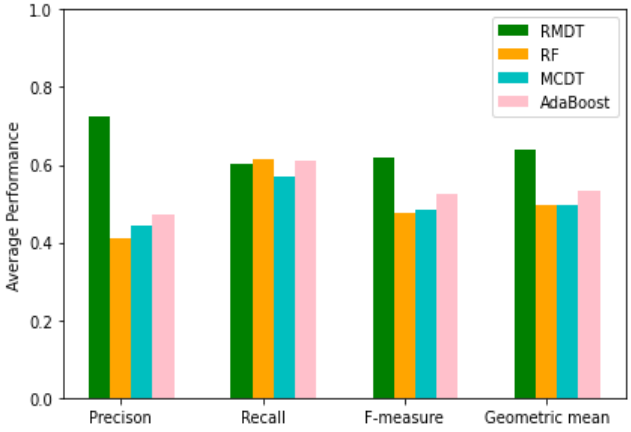


Fig.4: The experimental results on real-world datasets comparing by the average performance

Furthermore, for comparison by the recall, RMDT yields a similar average performance at 0.602 to RF, and AdaBoost but higher than MCDT. It means that the number of RMDT predicted minority instances to be the majority class is lower than MCDT but similar to RF, and AdaBoost. For comparison by the F-measure and the Geometric mean, they are not exhibiting the different results. RMDT yields the highest average performance at 0.620 and 0.638, which is better than other classifiers respectively.

5 Conclusion and Future Works

This paper proposed an enhanced random forest called RMDT which is a random forest that used both of the standard decision trees and the MCDT that successfully handles the class imbalanced problem, which arises from extending the ME concept. The improved performance to classify an imbalanced dataset of RMDT is shown by two collections of experiments which are experiments on synthetic binary-class imbalanced numeric datasets and real-world binary-class datasets from UCI, respectively. In the first experiment, RMDT outperforms RF when the number of minority instances decreases, and then their values will approach the same values when the dataset is more balanced. These apparently confirm that RF is not suitable for dealing with the binary-class imbalanced problem. Additionally, in the second experiment, RMDT performs better than RF, MCDT, and AdaBoost on the precision, the recall, the F-measure, and the Geometric mean. Especially, the precision of it shows the highest value which indicates that RMDT has high accuracy prediction.

Finally, although RMDT is successful in handling the class imbalanced problem, there is considerable room for future work. The proposed algorithm still has to be extended in order to function on more complex datasets, such as multi-class imbalanced datasets, and multi-class with categorical attributes imbalanced datasets. Additionally, the application of the proposed algorithm on EEG signals for Epilepsy Detection [13] is interesting for continued work. Including research recently [14], present a new multi-criteria decision making method that is intriguing to apply in the decision tree for the construction of RMDT.

References:

[1] A.Sagoolmuang and K. Sinapiromsaran, *Self-balancing recursive partitioning algorithm for classification problems*, 2019.

[2] BuczakAL, GuvenE, A survey of data mining and machine learning methods for cybersecurity intrusion detection, *IEEE Commun Surv Tutor*, 2016, pp. 1153–1176.

[3] A. Awad, M. Bader-El-Den, J. McNicholas, and J. Briggs, Early hospital mortality prediction of intensive care unit patients using an ensemble learning approach, *Int. J. Med. Information.*, Vol. 108, 2017, pp. 185–195.

[4] A. Fernánd z, S. García, M. Galar, R. C. Prati, B. Krawczyk, and F. Herrera, *Learning from imbalanced data sets*, Springer, 2018.

[5] K. Boonchuay, K. Sinapiromsaran, and C. Lursinsap, Decision tree induction based on minority entropy for the class imbalance problem, *Pattern Analysis and Applications*, Vol. 20, No. 3, pp. 769–782, 2017.

[6] Chandra B, Kothari R, Paul P, A new node splitting measure for decision tree construction, *Pattern Recognition*, Vol. 43, 2010, pp. 2725–2731.

[7] J. W. Tukey, *Exploratory data analysis*, Reading, Mass., Vol. 2, 1977.
[8] Thomas G. Dietterich, *An experimental comparison of three methods for constructing ensembles of decision trees*, Mach Learning, Vol.40, 2000, pp. 139–157.
[9] Buckland MK and Gey FC, *The relationship between recall and precision*, J Am Soc Info Sci, 1994, pp, 12–19
[10] C. L. Blake and C. J. Merz, Uci repository of machine learning databases, 1998.
[11] Scikit-learn, Machine Learning in Python, Pedregosa *et al.*, *JMLR*, Vol. 12, 2010, pp. 2825-2830.
[12] Bahad P. and Saxena P., Study of AdaBoost and Gradient Boosting Algorithms for Predictive Analytics, *International Conference on Intelligent Computing and Smart Communication*, 2019, pp. 235-244.
[13] Ayman M. Mansour, Mohammad A. Obeidat, Murad Al-Aqtash, Intelligent Classifiers of EEG Signals for Epilepsy Detection, *WSEAS Transactions on Signal Processing*, Vol. 15, 2019, pp. 106-113.
[14] Limin Su, Huishuang He, Hongwen Lu, Multi-criteria Decision Making Method with Interval Neutrosophic Setting based on Minimum and Maximum Operators, *International journal of circuits, systems and signal processing*, Vol. 13, 2019, pp. 177-182.