Improving ADABoost Algorithm with Weighted SVM for Imbalanced Data Classification

Vo Duc Quang[1,2(✉)], Tran Dinh Khang[1(✉)], and Nguyen Minh Huy[1]

[1] Hanoi University of Science and Technology, Hanoi, Vietnam
khangtd@soict.hust.edu.vn, huy.nm170773@sis.hust.edu.vn
[2] Vinh University, Vinh City, Vietnam
quangvd@vinhuni.edu.vn

Abstract. Recently, different boosting algorithms have been proposed in order to improve the performance of classification for imbalanced data. In this paper, we present an improved ADABoost algorithm, called Im.ADABoost, for imbalanced data including two main improvements: *(i)* initializing different error weights adapted to the imbalance rate of the datasets; *(ii)* calculating the confidence weights of the member classifier that is sensitive to the total errors caused on the positive label. Additionally, we combine Im.ADABoost with Weighted-SVM to enhance classification efficiency on imbalanced datasets. Our experimental results show some promising potential of the proposed algorithm.

Keywords: Imbalanced dataset · ADABoost · Support vector machine

1 Introduction

The analysis of imbalanced datasets has received much attention from researchers, especially in practical problems such as diagnosing diseases in medicine, detecting environmental problems, financial transaction fraud, and network attacks,... However, there are a number of the major challenge remains that need further study in the field of machine learning and data mining [12]. In this study, we consider classification problems on the imbalanced data of two class labels, in which the class label with the majority of data samples is called the negative label (-1) and the other with the minority of data sample is called the positive label $(+1)$.

Usually, traditional classification algorithms always consider data samples as equal. Algorithmic improvement studies are often geared towards trying to train the classification model with the highest accuracy rate. However, when applying these algorithms to imbalanced dataset problems, the classification model will be biased towards prioritizing class label recognition as (-1). In the case of a highly imbalanced dataset, the trained classification model tends to classify all label samples $(+1)$ as labels (-1). These results give a very high accuracy but fail to

T. K. Dang et al. (Eds.): FDSE 2021, LNCS 13076, pp. 125–136, 2021.
https://doi.org/10.1007/978-3-030-91387-8_9

properly classify any of the positive label samples. In the context of problems that need to accurately detect all labels (+1), this becomes meaningless. According to [10], approaches to improve classification on imbalanced datasets usually are:

- Using preprocessing methods on the dataset: (*i*) reduce the number of samples on the label (−1); (*ii*) generate additional artificial data samples label (+1) or (*iii*) combine both techniques. The above methods aim to reduce the imbalance of datasets in order to make traditional machine learning algorithms work efficiently [3,19,26].
- Improving algorithms: adjust the traditional algorithms so that they are more suitable for (+1) labels. The most popular method is error weight assignment, cost-based learning [1,5,9,14,18,21,22]. Some other studies apply deep learning models to imbalanced datasets [4,11,13,25].

Among the above approaches, the cost-based learning algorithm assigns a higher cost weight when the model misclassifies the sample label (+1) to the label (−1). Accordingly, this algorithm has many outstanding advantages such as: keeping the original characteristics of datasets, having many ways to improve the training parameters, and minimizing the error cost function by using loops in conjunction with turning the parameters. However, using a particular algorithm may not fully consider the attributes of datasets. Therefore, many studies combine member classifiers to produce a better composite classifier [7,8]. In particular, the ADABoost algorithm proposed by Freund [6] has been improved by many researchers, notably the study of combining ADABoost with Support Vector Machine (SVM) [2,15–17,20,23,24]. These improvements are intended to take advantage of ADABoost's adaptive iterability and SVM's scalability on datasets with different characteristics. In [15], the authors proposed a method that combines ADABoost with Weight SVM (W-SVM) to improve classification efficiency on imbalanced data.

However, the algorithm in [15] and other studies using ADABoost on imbalanced data initialize equal error weights for each data sample. This algorithm is not suitable when being applied to problems that need to prioritize the correct classification of labels (+1). In addition, ADABoost calculates the confidence weight of the membership classification algorithm based on the total error in the entire dataset without considering the details of each label type (+1) and (−1). Based on these observations, we propose an algorithm, called Im.ADABoost, by making two major improvements to the original ADABoost. Our improvements consist of: (*i*) initializing the set of different error weights adapted to the imbalance rate of the datasets; (*ii*) calculating the confidence weights of the member classifiers based on sensitivity to the total error caused on positive labels (+1), i.e., if the member classifier misclassifies more samples (+1), the lower its confidence weights will be.

We also combine Im.ADABoost with W-SVM into Im.ADABoost.W-SVM algorithm to classify imbalanced datasets. We used the recall, accuracy and fscore measures in our experiments to compare the performance with other classification algorithms on different imbalanced datasets. Experimental results show that

the Im.ADABoost.W-SVM algorithm gives better classification performance on imbalanced datasets, especially when datasets have a high imbalance ratio. The rest of the paper is structured as follows: Sect. 2 recalls the description of the original ADABoost algorithm and related ones; Sect. 3 presents improvements to the ADABoost algorithm combined with the W-SVM membership classifier; Sect. 4 describes the experimental results; Some discussion of the results and directions for future work are presented in Sect. 5.

2 Preliminaries

In classification problems on imbalanced datasets, using only a particular classification algorithm may not be efficient. ADABoost is an iterative algorithm that combines membership classification algorithms. This allows detailed testing of each sample in the dataset space by assigning each data sample an error weight. Through each iteration, the ADABoost re-evaluates the classification results of each membership classification algorithm, thereby calculating better parameters to use for the next iteration. The ADABoost algorithm is presented in Algorithm 1.

Algorithm 1: ADABoost

Input: A dataset with N samples $X = \{(x_1, y_1), \ldots, (x_N, y_N)\}$ with $y = \{-1, +1\}$, T: maximum iteration, h_i: the member classifier

Output: H: Ensemble classifier

1 **initialize:** the error weight $D^1 = \{\omega_i^1\}$ on each sample x_i with $i = 1, \ldots, N$

2 **for** $t = 1$ **to** T **do**

3 Set $h_t \leftarrow \text{Training}(X)$ with the error weight D^t;

4 Calculate total error traning: $\varepsilon_t = \sum_{t=1}^{N} \omega_i^t, y_t \neq h_t(x_i)$;

5 Calculate confident weight of h_t: $\alpha_t = \frac{1}{2} ln \frac{1-\varepsilon_t}{\varepsilon_t}$;

6 Update the error weight for the next iteration: $\omega_i^{t+1} = \frac{\omega_i^t . exp[-\alpha_t y_i h_t(x_i)]}{L_t}$, where L_t the normalization constant and $\sum_{i=1}^{N} \omega_i^{t+1} = 1$;

7 **return** $H = sign(\sum_{t=1}^{T} \alpha_t h_t)$.

The inputs of the algorithm include: (i) X is a dataset with N samples (x_i, y_i), where x_i is the attribute vector and y_i is the class label $y_i \in \{-1, +1\}$; (ii) D^1 is a set of equal error weights for each sample $w_i^t = \frac{1}{N}$; (iii) T is the maximum number of iterations; (iv) h_t is the membership classification algorithms.

In each loop, the classifier h_t classifies the dataset X at Step 3 of Algorithm 1. The classification quality of h_t is evaluated through the sum of error ε_t at Step 4 and confidence weight α_t at Step 5. Then the algorithm updates the error weight distribution ω_i^{t+1} using the formula at Step 6.

The synthetic classification model is calculated according to the formula H at Step 7. The classification label of the sample is determined based on the sign

function: label $(+1)$ when $H > 0$ and label (-1) when $H < 0$. When the total error ϵ_t on the dataset is equal to 0.5, then $\alpha_t = 0$, then the classifier h_t does not contribute to the decision of the composite classifier H.

We can see that the error weight ω_i^t assigned to each data sample is initialized equally. At each iteration, ADABoost analyzes the classification result of each member learner and updates the weights for each sample, as follows: increase the error weight assigned to the sample if it is misclassified and decrease the error weight if the sample is correctly classified. However, in the case of an imbalanced dataset, we need to adjust the algorithm to take a closer look at the positive labels, that is, to assign a higher error weight to the positive label samples. In addition, ADABoost calculates the confidence weight of the membership classification algorithm based on the total error on the entire dataset without considering the details of each label type $(+1)$ and (-1). These observations are the basis for us to develop ADABoost improvement methods for imbalanced data in Sect. 3, in which the algorithm prioritizes increasing the error weights when the member classifier misclassifies a positive label sample $(+1)$.

Wonji Lee et al. [15] proposed to combine ADABoost with W-SVM as a membership classification algorithm. This algorithm uses parameters z_i^t to adjust the weights on samples in W-SVM. The value of z_i^t is calculated based on the number of samples x_i distributed in the SVM marginal space. The formula to calculate z_i^t is :

If the sample x_i is in the border subclass BSV:

$$z_n^t = \begin{cases} \frac{N_{BSV}}{2N_{BSV-}} & if \quad y = -1, \\ \frac{N_{BSV}}{2N_{BSV+}} & if \quad y = +1. \end{cases} \tag{1}$$

If the sample x_i is in the border subclass SV:

$$z_n^t = \begin{cases} \frac{N_{SV}}{2N_{SV-}} & if \quad y = -1, \\ \frac{N_{SV}}{2N_{SV+}} & if \quad y = +1. \end{cases} \tag{2}$$

If the sample x_i is noise:

$$z_n^t = exp(\frac{N_{noisy}}{N_+}). \tag{3}$$

In the above formulas, N_{BSV} is the number of samples in the SVM boundary, N_{SV} is the number of samples generating the support vector and N_{Noisy} is the noise sample; Positive class label samples are represented by $(+)$, negative class label samples are represented by $(-)$; N_+ is the number of positive label samples. Based on the ADABoost.W-SVM algorithm, in Sect. 3, we propose two improvements to ADABoost algorithm, then combine our improved algorithm with W-SVM to classify the imbalanced datasets.

3 Proposed Method

3.1 Initialize Adaptive ADABoost Weights

In Sect. 2, we analyzed the limitation of ADABoost in initializing error weights when it is applied to an imbalanced dataset. In this section, we propose a new method to initialize error weights that adapts to the class-label imbalance ratio of the dataset. This method aims to assign a higher error weight on the positive label sample, +1. Assume N_{min} and N_{maj} are the number of samples of the minority and majority class, respectively, where $N_{min} \leq N_{maj}$. In ADABoost, each sample is assigned an error weight of $\frac{1}{N_{min}+N_{maj}}$ and therefore, ADBoost is inefficient for imbalanced datasets. We adjust the error weights by adding a Δ_{min} value to error weights of positive samples and subtracting a Δ_{maj} value from error weights of negative samples. This means the error weights on each positive sample will be $\frac{1}{N}+\Delta_{min}$, and on each negative sample will be $\frac{1}{N}-\Delta_{maj}$, where Δ_{min} and Δ_{maj} must satisfy the following conditions:

- Error weights are greater than 0, that is: $0 < \Delta_{min}, \Delta_{maj} < \frac{1}{N}$;
- The total error on the samples is equal to 1, that is:
  $\frac{N_{min}}{N} + N_{min} * \Delta_{min} + \frac{N_{maj}}{N} - N_{maj} * \Delta_{maj} = 1$;
- When the number of positive samples is equal to the number of negative samples, the dataset is balanced, then the error weight on each sample is equal to $\frac{1}{N}$.

If the ratio of the number of samples of positive to negative labels is set to $\delta = \frac{N_{min}}{N_{maj}}$, with $0 < \delta \leq 1$, then the above expression becomes:

$$\begin{cases} 0 < \Delta_{min}, \Delta_{maj} < \frac{1}{N} \\ \Delta_{maj} = \delta * \Delta_{min}. \end{cases} \tag{4}$$

We propose to choose $\Delta_{maj} = \frac{1-\delta}{N}$, thus $\Delta_{min} = \frac{1-\delta}{\delta * N}$. Accordingly, the set of bias weights is $D^t = \omega_i^t$ with $i = 1, 2, \ldots N$, and

$$\omega_i^t = \begin{bmatrix} \frac{1}{N} + \frac{1-\delta}{\delta * N}, & if \quad y_i = +1 \\ \frac{1}{N} - \frac{1-\delta}{N}, & if \quad y_i = -1. \end{bmatrix} \tag{5}$$

It can be seen that, when applying the formula (5) to datasets with different imbalance rates, the error weights on positive and negative labels will increase and decrease respectively depending on δ. When the dataset is balanced, meaning that $\delta = 1$, then Δ_{min} and $\Delta_{maj} = 0$, and therefore the initialization weights D^1 return to ADABoost's default (error weights on all samples are equal to $1/N$). In addition, in order to dynamically adjust the Δ_{min} and Δ_{maj} values according to the individual characteristics of the datasets, we propose a more general formula using the exponential parameter θ as follows:

$$\begin{aligned} \Delta_{maj} &= \frac{(1-\delta)^\theta}{N}, \\ \Delta_{min} &= \frac{(1-\delta)^\theta}{\delta * N}. \end{aligned} \tag{6}$$

For each particular dataset, we can find the most suitable exponential parameter value through the process of testing on a given set of values. This improvement makes ADABoost more generalizable on datasets with different imbalance rates. Moreover, if the algorithm uses a threshold that eliminates unnecessary membership classifiers, then it converges faster, meaning that it also reduces the number of iterations.

3.2 Positive Label Sensitive Confidence Weights of the Membership Classifier

At Step 5 of Algorithm 1, the confidence weight α_t of the member classifier h_t is calculated by a function that is inversely proportional to the total error ε_t. We find that this total error is considered equally across the misclassified samples. For the classification problem on the imbalanced dataset, the algorithm should give priority to assigning a high error weight when it misclassifies many positive-label samples (+1). We propose a new total error ε_t^* instead of ε_t, which is calculated by the total error of positive labels, denoted by ε_t^+, and that of negative labels, denoted by ε_t^-, i.e. $\varepsilon_t^* = \varepsilon_t^- + \varepsilon_t^+$ where $\varepsilon_t^+ = \sum_{i=1}^{N} \omega_i^t, y_i \neq h_t(x_i), y_i = +1$ and $\varepsilon_t^- = \sum_{i=1}^{N} \omega_i^t, y_i \neq h_t(x_i), y_i = -1$. Obviously, ε_t^* depends on ε_t^+ and ε_t^-, and if we want our model to classify precisely on positive labels, then we need to increase ε_t^+ and therefore, we redefine ε_t^* as follows:

$$\varepsilon_t^* = \varepsilon_t^- + \gamma * \varepsilon_t^+, \text{ subject to } \gamma > 1. \tag{7}$$

Since $0 < \varepsilon_t^- + \varepsilon_t^+ < 1$, we choose $\gamma = 2 - (\varepsilon_t^- + \varepsilon_t^+)$. Then, the confidence weight of the model is

$$\alpha_t^* = \frac{1}{2} ln \frac{1 - \varepsilon_t^*}{\varepsilon_t^*}. \tag{8}$$

Obviously, the total error value ε_t^* in (7) of the model increases with the total error on the positive label ε_t^+, resulting in the confidence weight value α_t^* being adjusted down accordingly. This means the algorithm will try to correctly classify as many positive-label data samples as possible.

3.3 Im.ADABoost.W-SVM Algorithm

We call ADABoost with two improvements in Sects. 3.1 and 3.2 an Im.ADABoost. Accordingly, we propose an algorithm using Im.ADABoost combined with W-SVM, called Im.ADABoost.W-SVM, where W-SVM is used as a member classifier. The Im.ADABoost.W-SVM scheme is described in Fig. 1 and Im.ADABoost.W-SVM algorithm is shown in Algorithm 2.

Our Im.ADABoost algorithm initializes the inputs $z_i^1 = 1$ and D^1, the set of adaptive error weights, calculated by (5) and (6) in Sect. 3.1 (i.e. the input in Fig. 1). The algorithm runs for T iterations and in each loop it performs as follows. First, the algorithm uses W-SVM to classify the dataset X by using the parameters z_i^1 and the error weights on the samples $D_t = \omega_i^t$ with $i = 1, 2, \ldots, N$ (i.e. Step W-SVM in Fig. 1). Then, the algorithm computes and updates the

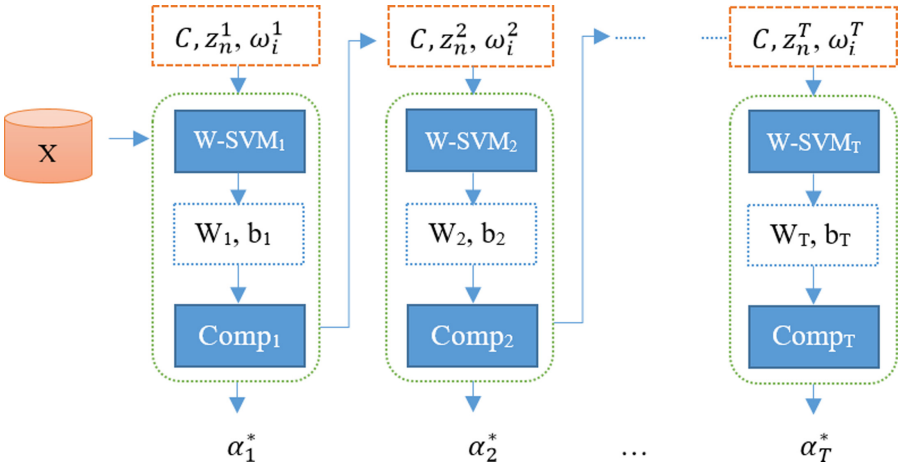

Fig. 1. Scheme of Im.ADABoost algorithm combined with W-SVM.

Algorithm 2: Im.ADABoost.W-SVM

Input: A dataset with N samples $X = (x_1, y_1), \ldots, (x_N, y_N)$, T: maximum iteration, h_i: the member classifier, C: W-SVM control parameters

Output: H: Ensemble classifier

1 **initialize**: $z_i^1 = 1, i = 1, 2, \ldots, N$ and ω_i^1 (using Eq.(5),(6));
2 **for** $t = 1$ **to** T **do**
3 $h_t \leftarrow$ Training W-SVM(X) with the error weight D^t and $z_i^t * \omega_i^t$, $i = 1, 2, \ldots, N$;
4 Calculate z_i^{t+1} (using Eq.(1),(2),(3));
5 Calculate total training error: ε_t^* (using Eq.(7));
6 Calculate the confidence weight of the h_t classifier: α_t^* (using Eq.(8));
7 Calculate error weight allocation for next loop: $\omega_i^{t+1} = \frac{\omega_i^t e^{-\alpha_t y_i h_t(x_i)}}{L_t}$, L_t is normalization constant, $\sum_{i=1}^{N} \omega_i^{t+1} = 1$;
8 **return** $H(x) = sign(\sum_{t=1}^{T} \alpha_t^* h_t(x))$.

parameters z_i^2 and D_2 for the next loop based on the $true/false$ classification results on the samples (i.e. Step Comp_1 in Fig. 1). It should be noted that the reliability α_t^* is calculated by (8) in Sect. 3.1. After the completion T loops, the aggregate classifier H predicts the class labels for the samples using the formula: $H(x) = sign(\sum_{t=1}^{T} \alpha_t^* h_t(x))$.

4 Experiments

In this section, we present experiments to evaluate the performance of our Im.ADABoost algorithm. To do so, we compared the classification results of Im.ADABoost.W-SVM with that of ADABoost.W-SVM [15] and ADABoost

Table 1. Description of datasets

Index	Name	#instances	#variables	Positive ratio(%)
1	Vertebral column	310	6	32.26
2	Indian liver patient	583	10	28.64

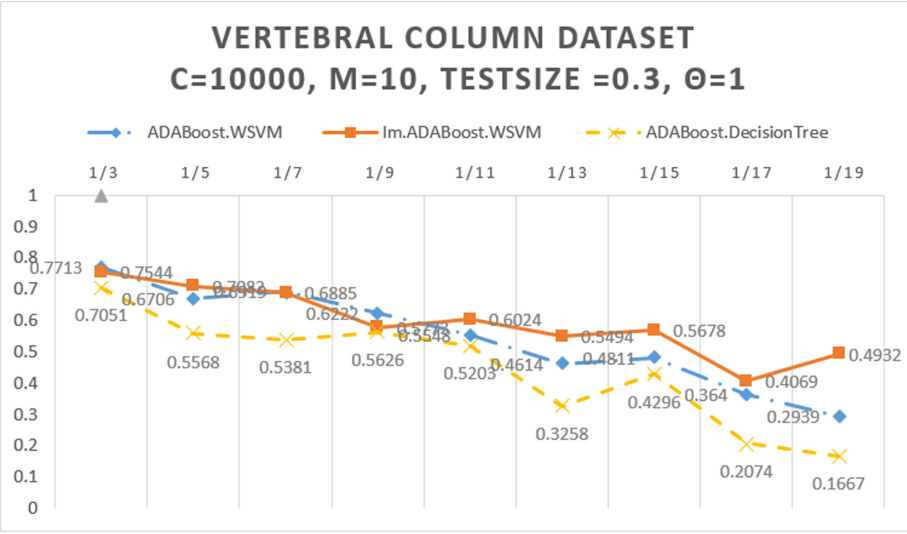

Fig. 2. Classification results for Vertebral column dataset.

DecisionTree [7] on two datasets with different imbalance rates[1] detailed in Table 1. On each dataset, we randomly select samples of (+1) labels, cut some positive labels (+1) to create a new sub-dataset with increasing imbalance ratio 1:5, 1:7, 1:9, 1:11, 1:13, 1:15, 1:17, 1:19. Our experimental scenarios use the following parameters: TestSize is 0.3 and 0.4; the number of Boosting iterations $T = 10$; the input parameter for SVM C is tested from 50:10:15000 to get the best value is $C = 10000$; the parameter θ of the Im.ADABoost is tested from 20:-1:1 to choose the most suitable for the dataset. We compare the classification efficiency of the algorithms by using the measures: *precision*, *recall*, *F1-Score*, taking the average results from the 10-fold method.

Using the *F1-Score* measure, $\theta = 1$, the classification results of positive labels (+1) of the algorithms on the *Dataset Vertebral column* dataset are presented in Fig. 2. The classification results of the Im.ADABoost.W-SVM algorithm on the *Dataset Vertebral column* dataset with different θ values are illustrated in Fig. 3. These results show that with datasets have positive label ratio, from 1:3 to 1:19, the ADABoost.W-SVM and Im.ADABoost.W-SVM algorithms always gives better results than ADABoost.DecisionTree. In

[1] https://www.kaggle.com/datasets.

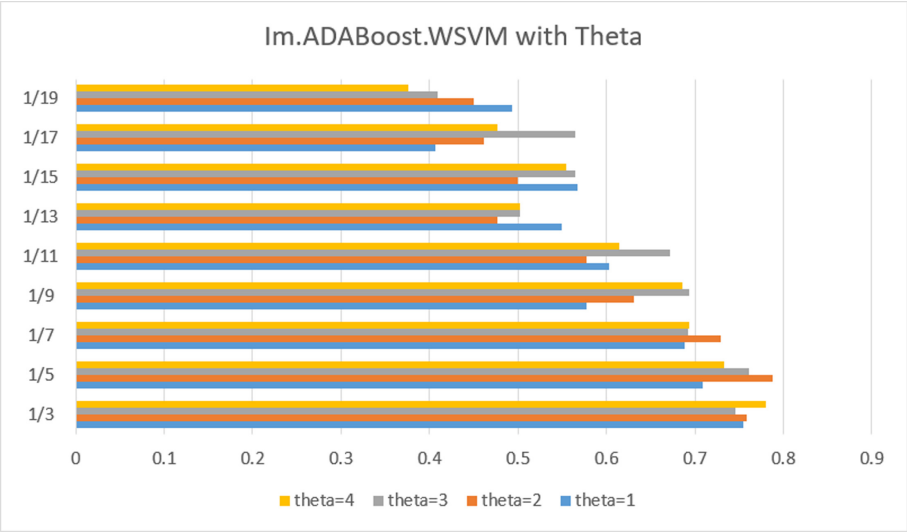

Fig. 3. Classification results for Vertebral column dataset with Theta.

addition, when the dataset has a low imbalance ratio, positive label ratio from 1:3 to 1:9, ADABoost.W-SVM and Im.ADABoost.W-SVM give approximately the same results. However, we can always find a suitable value θ that makes Im.ADABoost.W-SVM better than ADABoost.W-SVM. When the dataset has a high imbalance ratio, positive label ratio from 1:11 to 1:19, the Im.ADABoost.W-SVM algorithm gives a much better classification performance than ADABoost.W-SVM and ADABoost.DecisionTree with most experimental values θ.

Classification results on the *Indian liver patient* datasets are presented in Table 2. In all parameters used, these results show that the ADABoost.W-SVM algorithm does not correctly classify any positive label, and ADABoost.DecisionTree gives progressively worse results according to the decrease in the percentage of positive labels while Im.ADABoost.W-SVM correctly classifies all positive labels. This is very meaningful in the prediction problems of rare events. Moreover, when observing the values of θ, we see that, for different imbalance ratios of datasets, we can always choose a suitable θ value to help Im.ADABoost.W-SVM give the best performance. For example, given that dataset has a positive label ratio is 1:7, when testing the value $\theta = 20 : -1 : 1$, if $\theta \leq 4$, the Im.ADABoost.W-SVM algorithm gives results $recall = 1$, i.e. correctly classifies all positive labels $(+1)$. It should also be noted that the adjusted values for error weights in the Im.ADABoost algorithm decrease as the θ exponent increases accordingly. When $\theta > 4$, the adjusted value for the error weights of the samples is too small, and this impact is not enough to prioritize the correct classification of positive labels $(+1)$ in the Im.ADABoost algorithm.

Table 2. Classification results for Indian liver patient dataset.

Dataset **Indian liver patient**, C=10000, M=10, testsize =0.3, 0.4

positive label rate (+1)	Test size	ADABoost .WSVM				Im.ADABoost .WSVM			ADABoost .ID3		
		Precision	Recall	F1	θ	Precision	Recall	F1	Precision	Recall	F1
1:5	**0.3**	0.0000	0.0000	0.0000	**3**	0.1975	1.0000	0.5987	0.2852	0.1902	0.2377
	0.4	0.0000	0.0000	0.0000		0.2019	1.0000	0.6010	0.2749	0.1980	0.2365
1:7	**0.3**	0.0000	0.0000	0.0000	**4**	0.1438	1.0000	0.5719	0.2377	0.1328	0.1853
	0.4	0.0000	0.0000	0.0000		0.1443	1.0000	0.5722	0.2195	0.1447	0.1821
1:9	**0.3**	0.0000	0.0000	0.0000	**6**	0.1064	1.0000	0.5532	0.1816	0.0982	0.1399
	0.4	0.0000	0.0000	0.0000		0.1070	1.0000	0.5535	0.1748	0.1316	0.1532
1:11	**0.3**	0.0000	0.0000	0.0000	**7**	0.0870	1.0000	0.5435	0.1197	0.0746	0.0971
	0.4	0.0000	0.0000	0.0000		0.0870	1.0000	0.5435	0.0846	0.0592	0.0719
1:13	**0.3**	0.0000	0.0000	0.0000	**8**	0.0735	1.0000	0.5368	0.0704	0.0579	0.0642
	0.4	0.0000	0.0000	0.0000		0.0778	1.0000	0.5389	0.1130	0.0789	0.0960
1:15	**0.3**	0.0000	0.0000	0.0000	**10**	0.0672	1.0000	0.5336	0.1212	0.0585	0.0898
	0.4	0.0000	0.0000	0.0000		0.0674	1.0000	0.5337	0.1127	0.0746	0.0936
1:17	**0.3**	0.0000	0.0000	0.0000	**11**	0.0602	1.0000	0.5301	0.0301	0.0263	0.0282
	0.4	0.0000	0.0000	0.0000		0.0565	1.0000	0.5282	0.1066	0.0684	0.0875

5 Conclusion

In this paper, we have proposed an algorithm, called Im.ADABoost, by making two major improvements to the original ADABoost. The main modifications include *(i)* initializing the set of different error weights adapted to the imbalance rate of the datasets, which is adjusted by a parameter θ; and *(ii)* calculating the confidence weights of the member classifiers based on sensitivity to the total error caused on positive labels (+1), i.e., if the member classifier misclassifies more samples (+1), the lower its confidence weights will be. We also combine Im.ADABoost with W-SVM to classify the imbalanced datasets. Preliminary experimental results on two datasets named Vertebral Column and Indian Liver Patient show that Im.ADABoost achieves high classification efficiency compared to the original ADABoost algorithm with DecisionTree and ADABoost.W-SVM. Especially, when the datasets have small numbers of positive labels +1, the Im.ADABoost algorithm gives superior classification results. In addition, for each specific dataset, it is always possible to find the adjustable exponential parameter θ by experiment to help Im.ADABoost achieve the best classification results. We will further improve the calculation of confidence weights of the member classifier based on sensitivity total errors on positive labels and do more experiments on other datasets.

References

1. Akbani, R., Kwek, S., Japkowicz, N.: Applying support vector machines to imbalanced datasets. In: Boulicaut, J.-F., Esposito, F., Giannotti, F., Pedreschi, D. (eds.) ECML 2004. LNCS (LNAI), vol. 3201, pp. 39–50. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-30115-8_7
2. Benjamin, X.W., Nathalie, J.: Boosting support vector machines for imbalanced data sets. Knowl. Inf. Syst. **21**, 1–20 (2010). https://doi.org/10.1007/s10115-009-0198-y

3. Chawla, N., Bowyer, K., Hall, L., Kegelmeyer, W.: SMOTE: synthetic minority over-sampling technique. J. Artif. Intell. Res. **16**, 321–357 (2002)
4. Dong, X., Gao, H., Guo, L., Li, K., Duan, A.: Deep cost adaptive convolutional network: a classification method for imbalanced mechanical data. IEEE Access **8**, 71486–71496 (2020). https://doi.org/10.1109/ACCESS.2020.2986419
5. Elkan, C.: The foundations of cost-sensitive learning. In: Proceedings of the Seventeenth International Conference on Artificial Intelligence: 4–10 August 2001, Seattle, vol. 1, pp. 973–978 (2001)
6. Freund, Y.: Boosting a weak learning algorithm by majority. Inf. Comput. **121**(2), 256–285 (1995). https://doi.org/10.1006/inco.1995.1136
7. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. J. Comput. Syst. Sci. **55**(1), 119–139 (1997)
8. Galar, M., Fernández, A., Barrenechea, E., Bustince, H., Herrera, F.: Ordering-based pruning for improving the performance of ensembles of classifiers in the framework of imbalanced datasets. Inf. Sci. **354**, 178–196 (2016). https://doi.org/10.1016/j.ins.2016.02.056
9. Guo, H., Viktor, H.: Learning from imbalanced data sets with boosting and data generation: the databoost-im approach. SIGKDD Explor. **6**(1), 30–39 (2004). https://doi.org/10.1145/1007730.1007736
10. Hilario, A., Garcia Lopez, S., Galar, M., Prati, R., Krawczyk, B., Herrera, F.: Learning from Imbalanced Data Sets, Artificial Intelligence. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-98074-4_9
11. Johnson, J., Khoshgoftaar, T.: Survey on deep learning with class imbalance. J. Big Data **6**, 1–54 (2019). https://doi.org/10.1186/s40537-019-0192-5
12. Jordan, M., Mitchell, T.: Machine learning: trends, perspectives, and prospects. Science (New York N.Y.) **349**, 255–60 (2015). https://doi.org/10.1126/science.aaa8415
13. Khang, T.D., Tran, M.K., Fowler, M.: A novel semi-supervised fuzzy c-means clustering algorithm using multiple fuzzification coefficients. Algorithms **14**(9), 258 (2021)
14. Khang, T.D., Vuong, N.D., Tran, M.K., Fowler, M.: Fuzzy c-means clustering algorithm with multiple fuzzification coefficients. Algorithms **13**(7), 1–11 (2020)
15. Lee, W., Jun, C.H., Lee, J.S.: Instance categorization by support vector machines to adjust weights in AdaBoost for imbalanced data classification. Inf. Sci. **381**, 92–103 (2016). https://doi.org/10.1016/j.ins.2016.11.014
16. Li, X., Wang, L., Sung, E.: AdaBoost with SVM-based component classifiers. Eng. Appl. Artif. Intell. **21**, 785–795 (2008). https://doi.org/10.1016/j.engappai.2007.07.001
17. Lima, N.H.C., Neto, A.D.D., Dantas de Melo, J.: Creating an ensemble of diverse support vector machines using AdaBoost. In: 2009 International Joint Conference on Neural Networks, pp. 1802–1806 (2009)
18. Lin, C.F., Wang, S.D.: Fuzzy support vector machines. IEEE Trans. Neural Netw. **13**(2), 464–471 (2002). https://doi.org/10.1109/72.991432
19. Liu, X.Y., Wu, J., Zhou, Z.H.: Exploratory undersampling for class-imbalance learning. IEEE Trans. Syst. Man Cybern. Part B (Cybern.) **39**(2), 539–550 (2009). https://doi.org/10.1109/TSMCB.2008.2007853
20. Rengasamy, S., Punniyamoorthy, M.: Performance enhanced boosted SVM for imbalanced datasets. Appl. Soft Comput. **83**, 105601 (2019). https://doi.org/10.1016/j.asoc.2019.105601
21. Sun, Y., Kamel, M.S., Wong, A.K., Wang, Y.: Cost-sensitive boosting for classification of imbalanced data. Pattern Recogn. **40**(12), 3358–3378 (2007)

22. Tao, X., et al.: Affinity and class probability-based fuzzy support vector machine for imbalanced data sets. Neural Netw. **122**, 289–307 (2020)
23. Tharwat, A., Gabel, T.: Parameters optimization of support vector machines for imbalanced data using social ski driver algorithm. Neural Comput. Appl. **32**(11), 6925–6938 (2019). https://doi.org/10.1007/s00521-019-04159-z
24. Turki, T., Wei, Z.: Boosting support vector machines for cancer discrimination tasks. Comput. Biol. Med. **101**, 236–249 (2018). https://doi.org/10.1016/j.compbiomed.2018.08.006
25. Yan, Y., Chen, M., Shyu, M.L., Chen, S.C.: Deep learning for imbalanced multimedia data classification. In: 2015 IEEE International Symposium on Multimedia (ISM), pp. 483–488. IEEE, Miami (2015). https://doi.org/10.1109/ISM.2015.126
26. Zeng, M., Zou, B., Wei, F., Liu, X., Wang, L.: Effective prediction of three common diseases by combining SMOTE with Tomek links technique for imbalanced medical data. In: 2016 IEEE International Conference of Online Analysis and Computing Science (ICOACS), pp. 225–228 (2016). https://doi.org/10.1109/ICOACS.2016.7563084