A Novel Classification Method for Class-imbalanced Data and Its Application in microRNA Recognition

Xia Geng[1*], Yu-Quan Zhu[1], Zhi Yang[2]

[1]*School of Computer Science and Communication Engineering*
Jiangsu University
Zhenjiang, Jiangsu 212013, China
E-mails: gengxia@ujs.edu.cn, yuquanzhu@ujs.edu.cn

[2]*School of Management*
Jiangsu University
Zhenjiang, Jiangsu 212013, China
E-mail: yangzhi@ujs.edu.cn

[*]*Corresponding author*

Received: January 11, 2018 **Accepted: May 11, 2018**

Published: June 30, 2018

Abstract: *For non-coding RNA gene mining, especially microRNA mining, there are many challenges in the classification of imbalanced data. A novel classification method based on the Adaboost algorithm is proposed to handle the imbalance of positive and negative cases. Unstable-Adaboost is improved with respect to the initial weight assignment, the base classifier selection, the weight adjustment mechanism and other aspects. Furthermore, the Stable-Adaboost algorithm is proposed, which adjusts the weight of the sample set to rapidly achieve a more balanced training set. In addition, the Stable-Adaboost algorithm can ensure that the follow-up training set is maintained in a balanced state by optimizing the weight adjustment mechanism of incorrectly classified samples and stabilizing the classification performance. Experimental results show the superiority of Unstable-Adaboost and Stable-Adaboost in imbalance classification.*

Keywords: *Non-coding RNA, Class imbalance, Ensemble learning, Adaboost algorithm.*

Introduction

MicroRNA (miRNA) is a class of endogenous, noncoding, single-stranded RNA molecules that have a length of approximately 23 nt. Identification of miRNA and the prediction of corresponding target genes of miRNA are conducted to identify the biological functions and action mechanisms of miRNA. The accurate identification of miRNA can help researchers to analyze biological gene regulatory networks, understand the processes of post-transcriptional genes and guide drug development. Currently, research into the identification of miRNA is advancing, with the ab initio prediction method [5] being widely used. This method allows a classification model for unknown sequences to be established by extracting the secondary structure characteristics of miRNA precursor (pre-miRNA) molecules and combining the sequence features [18]. However, imbalanced classification is a serious problem of the method, as most of the positive examples are selected from experimental verification, whereas the negative ones typically are not. Therefore, negative examples are obtained with low cost and positive examples with high cost such that negative examples are typically far more abundant than positive examples in the training set. This problem is often encountered in SNP discrimination [12] and microArray data analysis [8].

Classification imbalance can occur in many areas [13, 20, 21], especially in binary classification problems, such as financial fraud detection [15], oil exploration [7] and anti-

spam [2]. However, the ordinary classification method of machine learning cannot be directly applied to these areas [19]. To solve the problems of learning imbalanced classification, a random sampling approach has been proposed, in which the training set of samples is changed so that equilibrium can be reached. The simplest methods of random sampling are over-sampling and under-sampling. Research has shown that the random over-sampling method typically results in problems such as long computation times and over-fitting; as a result, the random down-sampling method is more commonly adopted. However, the random down-sampling method only uses a subset of the majority class, precluding full use of the existing information. Recently, several manual sampling approaches have been proposed. SMOTE [1] is based on over-sampling but increases the minority class samples through artificial means rather than through random selection and copying, thereby avoiding the over-fitting problem, although noise can still be generated.

In addition to these sampling approaches, other methods have been applied to handle class-imbalanced data, such as the boosting method of ensemble learning [4], the cost-sensitive learning algorithm [25], the single-class learning method [11], cascade neural networks [10], and clustering and support vector machines [9]. Theoretical analysis and experimental data have demonstrated that among these other methods, the ensemble learning method produces the most satisfactory results with class-imbalanced datasets. This method combines some weak classifiers into one strong one with high accuracy (ensemble classifier). Thus, compared with the original classification model, this method can improve the classification accuracy of the minority class. Adaboost [3] is a well-known ensemble learning algorithm that can effectively improve the generalization ability of the base classifiers. In recent years, some improved algorithms for processing imbalanced data have emerged, such as the RareBoost algorithm, the Cost-Boosting algorithm, and the AdaCost algorithm, which have better learning effects. To address the problem of imbalanced classification in miRNA identification, an integrated algorithm LibID was proposed by Zou et al. [26]. LibID employs a strategy similar to Adaboost, and it can yield a better recognition effect while ensuring sensitivity and improving specificity. However, it uses different base classifiers and a repeated training sample, resulting in a slightly longer training time compared with that of the general integration algorithm. The algorithm PlantMiRNAPred, proposed by Xuan et al. [22], also employs a strategy similar to Adaboost, but it considers the extreme imbalance factors between the positive and negative examples. It uses filtering steps to filter the samples that are easy or difficult to classify, and new classifiers are used to handle the samples that are readily classified incorrectly. However, the PlantMiRNAPred algorithm is mainly used to predict true and false plant pre-miRNAs.

Several research methods for miRNA prediction have appeared in recent years, such as the method proposed by Kamarajan et al. [6] and one proposed by Wang [17]. Extensive research on the classification imbalance problem in miRNA recognition has been conducted, and some related algorithms have been proposed. A comprehensive comparison of the efficiency and performance of these algorithms has been conducted in [16]. A difficulty encountered in developing such methods is how to select a representative sample from the positive and negative class-imbalance data to adequately describe the whole sample space. The algorithm MatFind is proposed by Ying et al. [24]. They proposed that the ensemble SVM classifiers and balanced-datasets can solve the class-imbalanced problem, as well as improve performance of classifier for mature miRNA identification. MatFind is an accurate and fast method for 5' mature miRNA identification. The algorithm ELM, proposed by Rodriguez et al. [14], is a novel approach to overcome the high class imbalance in pre-miRNAs prediction data in which ELMs are used for predicting good candidates to pre-miRNA, without needing

balanced data sets. The present paper investigates the processing imbalance problem of the Adaboost algorithm in pre-miRNA identification and improves and optimizes the Adaboost algorithm with respect to classification performance, efficiency and stability. Experimental results show that this method can effectively improve the performance of the classifier in imbalanced data.

Weight-adjustment optimization mechanism of the stable-Adaboost algorithm

The Adaboost algorithm has some problems in imbalanced dataset classification.

Problem 1. The algorithm requires more cycles to complete the entire weight-adjustment process, and the classifiers for the majority class that are produced in this training process are far more abundant than are those for the minority class.

Problem 2. The balanced dataset obtained by extracting from the training set cannot remain stable; the dataset will become imbalanced again in the process of the algorithm.

The Adaboost algorithm can be improved with respect to these problems. By optimizing the initial sampling weights during the weighted random-sampling process of the Adaboost algorithm to rapidly balance the number of each class in the training set, classifiers with high accuracy for each class can be achieved. Then, the problem of how to assign the weights is related to the mathematical probability.

Assuming that the dataset contains minority class *A* and majority class *B*, that the probability that each sample in A can be extracted is the same, and that the probability that each sample in B can be extracted is the same, then the problem of sample balance is converted to an equal probability question.

Suppose a random sample *s* is extracted from the dataset containing *A* and *B* and the probability of *s* being extracted from *A* is same as the probability of being extracted from *B*, then the initial weights of the minority class and majority class can be calculated by the following formula:

$$W_A \times N_A - W_B \times N_B = 0. \tag{1}$$

Therefore, the initial weight ratio of the two categories of samples can be calculated by the formula $W_A / W_B = N_B / N_A$.

By adjusting the initial weights, *Problem 1* is solved, and *Problem 2* is solved by the weight adjustment. In the Adaboost algorithm, the weight adjustment method after training is as below.

$$Weight_{new} = Weight_{current} \times \frac{Er(C_i)}{1 - Er(C_i)}, \left(Er(C_i) < 0.5\right), \tag{2}$$

where $Er(C_i) = Number_{wrongly\ classified\ samples} / Number_{training\ samples}$.

It can be found that when the base classifiers meet the conditions of the ensemble learning (that is, when the error rate $Er(C_i)$ is less than 0.5), the lower the error rate is, the more the weight of the correctly classified sample is reduced. This method ensures the comprehensive learning of the dataset; however, some problems remain for classifying imbalanced data. Therefore, we adjust the sample weights of the majority class by referring to the classical Adaboost algorithm after training while calculating the total weight reduction value of the samples in majority class *TRW*, and this value is equally distributed to each sample that is classified correctly in the minority class. This approach not only retains the comprehensive learning characteristics of the Adaboost algorithm for classification but also ensures the stability of the dataset and maintains a balanced dataset for further training.

In the selection process of the base classifiers, the error rate $Er(C_i)$ is adopted as the eligibility criteria of the base classifiers. The calculation method of the Adaboost algorithm is not used here because we found in experiments that when the number of iterations increases, the weight of the sample becomes smaller; therefore, the error rate according to the calculation method of the Adaboost algorithm cannot truly reflect the performance of the base classifiers. The weight calculation Eqs. (2)-(3) for the base classifier ensemble is the same as for the Adaboost algorithm.

$$Predict(x) = vote\{W_i \times C_i(x)\},\ i \in \{1, 2, 3, \cdots, t\}. \tag{3}$$

The following is the description of the improved Adaboost algorithm:

Input: weak classifier SVM (support vector machine), training dataset of Adaboost, the number of the base classifiers N (cycle number), the training sample number k;

Output: ensemble classifier $C = \{C_1, C_2, \cdots, C_N\}$;

The training phases:

(1) initialize the weight of each sample in Adaboost to achieve the balance of positive samples and negative samples in random sampling:

$$Weight_{negative\ sample} = Weight_{positive\ sample} / \left(number_{negative\ sample} / number_{positive\ sample} \right)$$

(2) for I = 1 to N do

(3) get training subset S_i by sampling with replacement from S according to the weight of the sample

(4) train the base classifier C_i by S_i and L

(5) calculate the error rate $Er(C_i)$:

$$Er(C_i) = Number_{wrongly\ classified\ samples} / Number_{training\ samples}$$

(6) if the $Er(C_i) > 0.5$ then

(7) initialize the weight of each sample in Adaboost again to achieve the balance of positive samples and negative samples in random sampling

(8) return to step (3) and try again

(9) end if

(10) for each majority sample correctly classified in S_i do

(11) $Weight_{new} = Weight_{current} \times (Er(C_i) / (1 - Er(C_i))$

(12) calculate total weight reduction value TRW

(13) end for

(14) for each correct classification of the minority class in the Si sample do

(15) $Weight_{sample} = TRW / Number_{correct\ classification\ sample\ in\ the\ minority\ class}$

(16) end for

(17) end for

The classification phases:

(18) input sample test;

(19) for I = 1 to N do

(20) $Weight_i = \log\left(\frac{1 - Er(C_i)}{Er(C_i)}\right)$

(21) end /* calculate the weight of each classifier */

(22) use N base classifiers to classify for a given sample x, and return classification results by the weighted vote combined method.

The experimental results show that the improved Adaboost algorithm can rapidly obtain sustainable training datasets with two balanced categories, achieve base classifiers with high classification accuracy for each class, and improve the classification accuracy by ensemble learning. The newly proposed algorithm reduces the processing time and improves the operational efficiency compared with the original algorithm, and it has high applicability for class-imbalanced data.

Experiment and results

Experimental dataset

In the experiment, we used a dataset of miRNA. MiRNA, also known as small RNA, refers to single-stranded, non-coding, regulatory RNA molecules of short length. Different from RNA,

miRNA is not translated into a protein when transcribed but plays a regulatory role in metabolism; thus, it is of great significance for research on biological genes.

The dataset used is the same one used in Xue et al. [23]. As a typical imbalanced class dataset, it includes 193 positive samples and 8494 negative samples, the ratio of which is 1:44. Each sample contains 32 characteristic attributes and one class attribute, and all of the attributes are numeric. The test set contain 30 positive samples and 1000 negative samples, and the training set is achieved by the improved Adaboost algorithm.

Experimental methods

Because the samples in the dataset contain many properties, SVM is selected as the classification algorithm. When dealing with binary-class problems and when the properties of the samples are continuous in time, SVM typically has better learning effects than do other learning algorithms. To accurately measure the performance of the classifier, some samples are reserved as test datasets for the ensemble classifier performance test before classification training. The ratio of positive and negative samples in the test dataset should be close to that of the original dataset (1:44).

In terms of evaluation criteria, sensitivity (*sn*) and specificity (*sp*), which have better measurement ability where biological class-imbalance data are concerned, as well as *gm* proposed by Wei et al. [18], are adopted to directly compare the overall performance of the classifiers.

TP denotes the number of positive samples that are correctly predicted, and *TN* denotes the number of negative examples that are correctly predicted. *FN* denotes the number of positive samples that are incorrectly predicted, and *FP* denotes the number of negative examples that are incorrectly predicted:

$$sn = TP / (TP + FN), \tag{4}$$

$$sp = TN / (FP + TN), \tag{5}$$

$$gm = sqrt(sn \times sp). \tag{6}$$

The experimental procedures are described below:

1. The ensemble classifier based on Adaboost is achieved, and the changes in the training set and the classification results are observed.

2. Adaboost is improved by modifying the initial weights and error rate calculation method (called the Unstable-Adaboost algorithm), the ensemble classifier based on this improvement is achieved and the training set changes and classification results are observed.

3. Adaboost is improved by modifying the initial weights and error rate calculation method and by weight adjustment after training (called the Stable-Adaboost algorithm). The ensemble classifier based on this improvement is achieved, and the training set changes and classification results are observed.

Unstable-Adaboost is actually a partially optimized method of Adaboost in terms of imbalanced data classification. This method adjusts the initial weights but avoids the lengthy first weight adjustment process, and it removes a large number of severely biased classifiers produced by this process. Afterwards, in the subsequent training, the balance degree of the training set remains volatile, which has a further impact on the ensemble classification.

Stable-Adaboost, based on Unstable-Adaboost, improves the weight adjustment after training, and it inhibits the structure volatility of the training set so that it can maintain the training set in an equilibrium state.

Analysis of experimental results

1. We set different iterations to train the classifier by Adaboost and observe the structural changes of the training set and the number of iterations when the first equilibrium is reached.

When the ratio of the number of positive to negative samples is 1:44, the Adaboost algorithm requires approximately 40 iterations to first obtain the training set with a ratio of the number of positive to negative samples close to 1:1. Therefore, this algorithm requires a long time to run, and the time requirement will increase with an increasing sample base and an increasing imbalance degree of the dataset, as shown in Fig. 1.

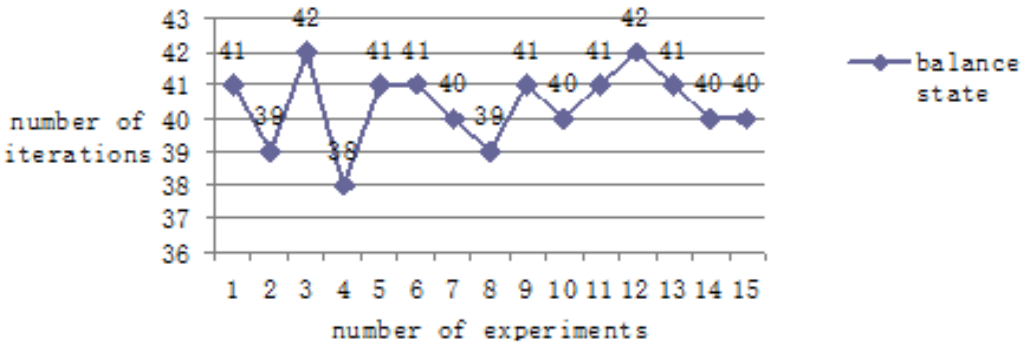

Fig. 1 Number of iterations required to stabilize the training set

2. We train the classifier by Unstable-Adaboost and observe the changes in the proportions of the positive and negative samples in the training set and the ensemble classifier performance under different numbers of iterations.

Fig. 2 shows that the training set extracted first is more balanced and that the training set has a large change in the sample proportion at the third extraction, with the ratio of the number of positive to negative samples decreasing to approximately 4:10. Thus, the changes in the ratio are less severe, with an overall tendency for the number of positive samples to increase and the number of negative samples to decrease. After several iterations, the extracted training set is balanced once again, followed by imbalance after additional training, after which the process is repeated.

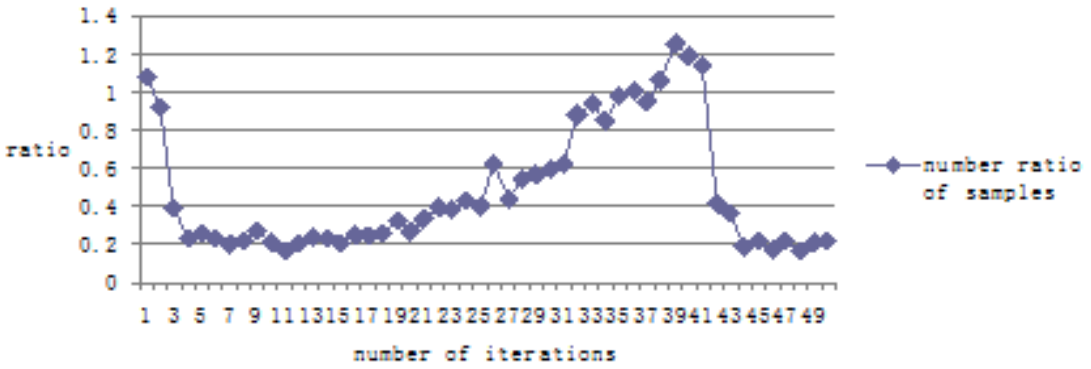

Fig. 2 The changing structure of the training set in the Unstable-Adaboost process

In Fig. 3, due to the adjustment of the initial weight, the training set extracted in the first training is balanced, and the classifiers trained have high overall performance *gm*. After this point, *gm* appears volatile. It can be seen in Fig. 2 that the imbalance degree of the training set increases sharply at the third training. Over approximately 30 trainings, the training set is always in an imbalanced state, and the performance of the ensemble classifier is reduced. From the 30th to 40th training, the training set returns to a balanced state, and the performance of the ensemble classifier increases. A second peak in *gm* occurs at the 40th training but is significantly weaker than before. After the 40th training, the training set appears imbalanced again, the ensemble classifier performance decreases, and Unstable-Adaboost enters the second transition process.

The ultimate goals of the ensemble classification are to improve the classification performance and to obtain ensemble classifiers with an overall performance that is better than the performance of the base classifiers. The performance of the ensemble classifier should exhibit an upward trend and then level off with an increasing number of training iterations. Unstable-Adaboost improves the classification performance by modifying the initial weights in Adaboost to reduce the redundancy; however, it is obvious that some problems remain.

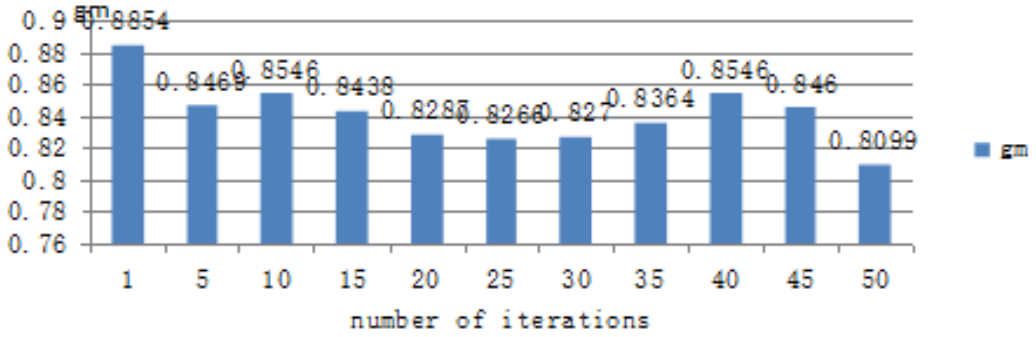

Fig. 3 *gm* of the ensemble classifiers in Unstable-Adaboost under different iterations

3. With different iterations, ensemble classifications are performed by Stable-Adaboost, during which the structure of the training set and integrated classifier performance changes are observed. The ratio of the positive and negative samples in the training set are shown in Fig. 4.

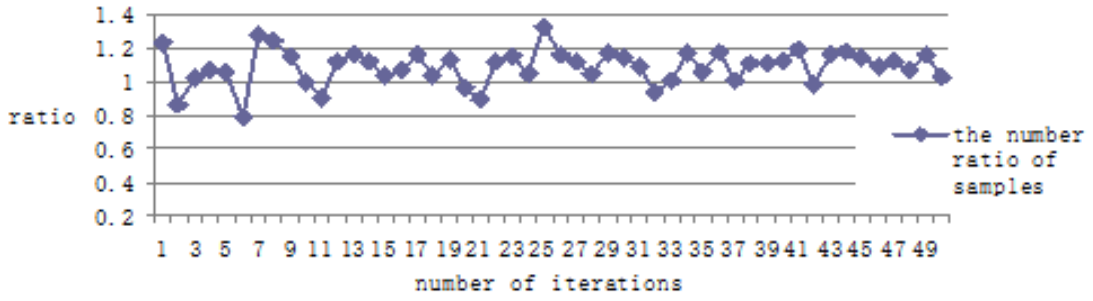

Fig. 4 The ratio of positive to negative samples in the training set during the iteration process

It shows that after the weight adjustment mechanism is modified, the training sets maintains a balanced state.

Fig. 5 shows that the performance of the ensemble classifier first increases with an increasing number of iterations and then tends to be stable. After several iterations, gm becomes stable in the vicinity of 0.914, and the performance improvement compared with that of a single classifier is obvious.

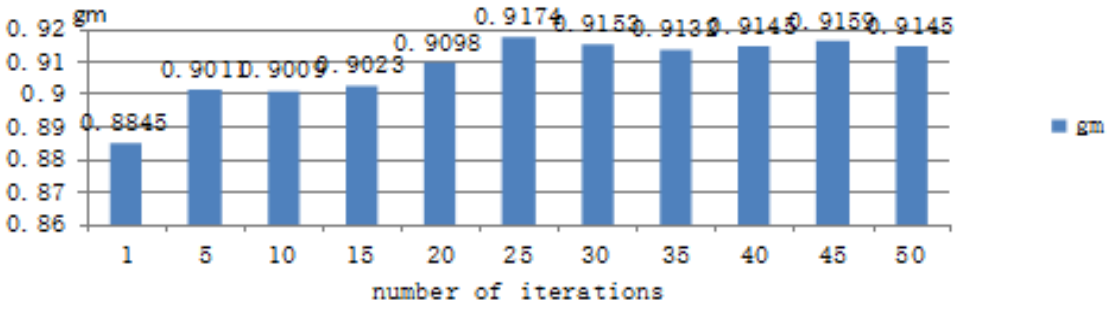

Fig. 5 *gm* of the ensemble classifiers in Stable-Adaboost under different iterations

Fig. 6 shows that Stable-Adaboost is much better than Unstable-Adaboost in sensitivity and that the recognition rate for the minority class increases significantly with Stable-Adaboost. The specificity of Stable-Adaboost is slightly lower than that of Unstable-Adaboost, which has almost the same recognition abilities for the majority class. Compared with Unstable-Adaboost, the overall performance of Stable-Adaboost is greatlyenhanced.

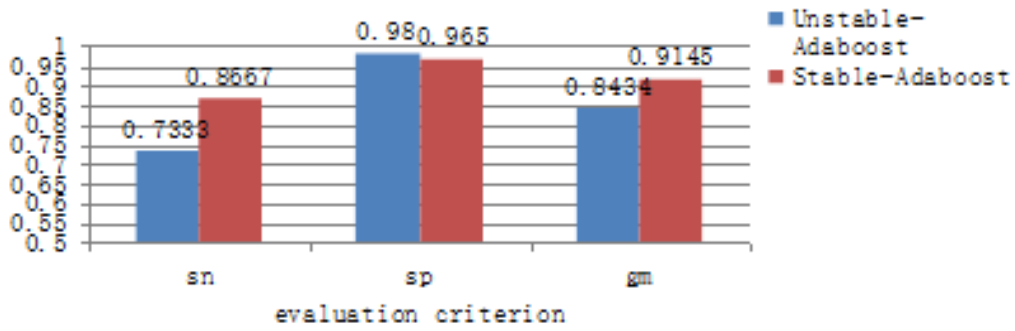

Fig. 6 The performance of ensemble classifiers in Unstable-Adaboost and Stable-Adaboost

UCI data

The UCI test datasets *cmc*, *haberman*, *ionosphere*, *letter* and *pima* are selected. These datasets have real number properties and are class imbalanced (in multi-class problems, the smallest class is considered as positive class and the remainder are seen as negative class). The methods contrasted with our methods are Adaboost (the base classifier is the decision tree method), random down-sampling (UnderSampl), mixed sampling (HSampl), AsymBoost and BalanceCascade, for which the experimental results from UCI test datasets are obtained from the literature. In addition, to verify the effectiveness of the proposed algorithm, we also compare Unstable-Adaboost and Stable-Adaboost. The experimental results are shown in Table 1.

Table 1. The results of the classifiers using the UCI dataset

Data (| P |/| N|)	Classifier	Precision	Recall
cmc (333/1140)	Adaboost	0.40	0.39
	UnderSampl	0.33	0.63
	HSampl	0.37	0.48
	AsymBoost	0.39	0.42
	BalanceCascade	0.35	0.59
	Unstable-Adaboost	0.42	0.62
	Stable-Adaboost	0.53	0.68
haberman (81/225)	Adaboost	0.35	0.36
	UnderSampl	0.36	0.60
	HSampl	0.36	0.47
	AsymBoost	0.34	0.39
	BalanceCascade	0.36	0.57
	Unstable-Adaboost	0.48	0.79
	Stable-Adaboost	0.56	0.86
ionosphere (126/225)	Adaboost	0.95	0.88
	UnderSampl	0.92	0.89
	HSampl	0.94	0.86
	AsymBoost	0.95	0.88
	BalanceCascade	0.93	0.89
	Unstable-Adaboost	0.93	0.88
	Stable-Adaboost	0.94	0.90
pima (268/500)	Adaboost	0.63	0.60
	UnderSampl	0.58	0.73
	HSampl	0.62	0.65
	AsymBoost	0.63	0.61
	BalanceCascade	0.60	0.71
	Unstable-Adaboost	0.77	0.74
	Stable-Adaboost	0.79	0.82
letter (789/19211)	Adaboost	0.99	0.98
	UnderSampl	0.83	0.99
	HSampl	0.92	0.99
	AsymBoost	0.99	0.98
	BalanceCascade	0.96	0.99
	Unstable-Adaboost	0.89	0.98
	Stable-Adaboost	0.85	0.98

As shown in Table 1, the methods proposed here have slightly worse performance than those of the previous methods only for the dataset *letter*, and their performances for the other datasets are superior to those of the other classification methods. Our methods are based on the ensemble learning theory; therefore, they are better suited to handle datasets with weak classification (e.g., *cmc* and *haberman*). *letter* is a strong classification dataset, and the decision tree based on Adaboost was almost completely accurate; the methods proposed here were less so. However, in general and particularly when dealing with a class-imbalanced dataset with weak classification, the improved methods proposed here show strong advantages.

The performance of Stable-Adaboost is better than Unstable-Adaboost on the datasets *haberman*, *ionospher*, *pima* and *cmc* and slightly worse than Unstable-Adaboost on the dataset *letter*. Due to the facts that *letter* is a strong classification dataset and that misclassified data of each classifier are very few, iteration training on *letter* is ineffective. Therefore, Stable-Adaboost yields a greater overall performance improvement than does Unstable-Adaboost.

Comparisons with similar algorithms

To verify the effectiveness of the proposed method, we compare the improved method to other similar algorithms by using a miRNA dataset. Xue et al. [23] studied the miRNA of human precursors and provided a dataset with 193 positive samples and 8494 negative samples, from which 163 positive samples and 168 negative samples were extracted as the training dataset by LibSVM random down-sampling, whereas 30 positive samples and 1000 negative sample were extracted as the test dataset. Zou et al. [26] adopted the same test dataset [22], but all of the samples that were not in the training set constituted the test dataset (163 positive samples and 7494 negative samples). Here, we adopt the same dataset, but the training dataset is extracted by weight, and the test dataset includes 30 positive samples and 1000 negative samples. Triplet-SVM was proposed by Xue et al. [23], and LibID was proposed by Zou et al. [26]. The comparison results are shown in Fig. 7.

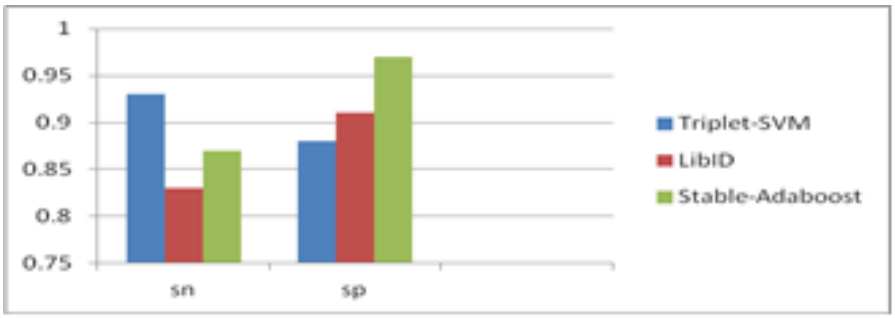

Fig. 7 Comparison results of similar algorithms

The comparison results show that the method we proposed considers more heavily the information of the negative samples such that the indicator *sp* value is higher for this method than for the other two methods. The indicator *sn* value is higher for Triplet-SVM because the positive samples in the training dataset are much more abundant than those in the test dataset; therefore, the over-fitting problem occurs. This problem is also mentioned in the Xue et al. [23], in which the indicator *sn* decreases when the same training dataset is used to predict other species. In addition, the improved method proposed can increase the indicator *sp* value compared with the values of the other two methods; meanwhile, the indicator *sn* is ensured, which is very important to molecular biology researchers.

Conclusion

To handle the sample class imbalance problem in bioinformatics, an improved method, Stable-Adaboost, is proposed, which is based on Adaboost. This method is improved with respect to the initial weight assignment, the base classifier selection, the weight adjustment mechanism and other aspects. Furthermore, it adjusts the weight of the sample set to rapidly achieve a more balanced training set. In addition, it can ensure that the follow-up training set is maintained in a balanced state by optimizing the weight adjustment mechanism of incorrectly classified samples and stabilizing the classification performance. Stable-Adaboost can eliminate the redundancy of the transition phase, maintain the stability of the training dataset and improve the performance of ensemble classification. In bioinformatics research, specificity is often more important than sensitivity because high specificity can reduce the cost of experimental verification. The method proposed here ensures sensitivity while increasing specificity in non-coding RNA class prediction.

References

1. Chawla N. V., K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer (2002). Smote: Syntheticminority Over-sampling Technique, Journal of Artificial Intelligence Research, 16, 321-357.
2. Fawcett T. (2003). *In vivo* Spam Filtering: A Challenge Problem for Data Mining, ACM SigKDD Explorations, 5(2), 140-148.
3. Freund Y., R. E. Schapire (1997). A Decision-theoretic Generalization of On-line Learning and an Application to Boosting, Journal of Computer and System Sciences, 55(1), 119-139.
4. Guo H., H. L. Viktor (2004). Learning from Imbalanced Data Sets with Boosting and Data Generation: The DataBoost-IM Approach, ACM SigKDD Explorations, 6(1), 30-39.
5. Hu L. L., Y. Huang, Q. C. Wang, Q. Zou, Y. Jiang (2012). Benchmark Comparison of *ab initio* microRNA Identification Methods and Software, Genetics and Molecular Research, 11(4), 4525-4538.
6. Kamarajan B. P., J. Sridhar, S. Subramanian (2012). *In silico* Prediction of microRNAs in Plant Mitochondria, International Journal Bioautomation, 16(4), 251-262.
7. Kubat M. S., R. C. S. Holte, S. S. Matwin (1998). Machine Learning for the Detection of Oil Spills in Satellite Radar Images, Machine Learning, 30(2), 195-215.
8. Li J. Z., K. Yang, H. Gao, J. Z. Luo, Z. Guo (2006). Model Free Genes Election Method by Considering Unbalanced Samples, Journal of Software, 17(7), 1485-1493.
9. Li P., X. L. Wang, Y. C. Liu, B. X. Wang (2007). A Classification Method for Imbalance Data Set Based on Hybrid Strategy, Acta Electronica Sinica, 35(11), 2161-2165.
10. Liu X. Y., J. X..Wu, Z. H. Zhou (2006). A Cascade-based Classification Method for Class-imbalanced Data, Journal of Nanjing University: Natural Sciences, 42(2), 148-155 (in Chinese).
11. Manevitz L. M., M. Yousef (2001). One-class SVMS for Document Classification, Journal of Machine Learning Research, 2(2), 139-154.
12. Marth G. T., I. Korf, M. D. Yandell, R. T. Yeh, Z. Gu, H. Zakeri, N. O. Stitziel, L. Hillier, P. Y. Kwok, W. R. Gish (1999). A General Approach to Single Nucleotide Polymorphism Discovery, Nature Genetics, 23(4), 452-456.
13. Pearson R., G. Goney, J. Shwaber (2003). Imbalanced Clustering for Microarray Time-series, Proceedings of the ICML'03 Workshop on Learning from Imbalanced Data Sets, Washington DC, http://www.elg.uottawa.ca/~nat/Workshop2003/pearson.pdf
14. Rodriguez T., L. E. D. Persia, D. H. Milone, G. Stegmayer (2017). Extreme Learning Machine Prediction under High Class Imbalance in Bioinformatics, Computer Conference. IEEE, 1-8.

15. Stolfo S., W. Fan, W. Lee, A. Prodromidis, P. Chan (1999). Cost-based Modeling for Fraudand Intrusion Detection: Results from the Jam Project, Proceedings of the 5th ACM SigKDD International Conference on Knowledge Discovery and Data Mining, SanDiego, 1-15.
16. Tran V., T. Du, S. Tempel, B. Zerath, F. Zehraoui, F. Tahi (2015). miRBoost: Boosting Support Vector Machines for microRNA Precursor Classification, Bioinformatics, 21, 775-785.
17. Wang C. (2015). A Modified Machine Learning Method Used in Protein Prediction in Bioinformatics, International Journal Bioautomation, 19(1), 85-96.
18. Wei L. Y., M. H. Liao, Y. Gao, R. R. Ji, Z. Y. He, Q. Zou (2013). Improved and Promising Identification of Human microRNAs by Incorporating a High-quality Negative Set, IEEE/ACM Transactions on Computational Biology and Bioinformatics, 11(1), 192-201.
19. Weiss G. M., W. R. Mining (2004). A Unifying Framework, ACM SigKDD Explorations, 6(1), 7-19.
20. Wu G., E. Y. Chang (2003). Class-boundary Alignment for Imbalanced Dataset Learning, Proceedings of the ICML'03 Workshop on Learning from Imbalanced Data Sets, Washington DC, 1-8.
21. Wu J., M. D. Mullin, J. M. Rehg (2005). Linear Asymmetric Classifier for Cascade Detectors, Proceedings of the 22nd International Conference on Machine Learning, 993-1000.
22. Xuan P., M. Z. Guo, X. Y. Liu, Y. C. Huang, W. B. Li, Y. F. Huang (2011). Plant MiRNAPred: Efficient Classification of Real and Pseudo Plant pre-miRNAs, Bioinformatics, 27(10), 1368-1376.
23. Xue C. H., F. Li, T. He, G. P. Liu, Y. D. Li, X. G. Zhang (2005). Classification of Real and Pseudo microRNA Precursors Using Local Structure-sequence Features and Support Vector Machine, BMC Bioinformatics, 6, 310.
24. Ying W., X. Li, B. Tao (2016). Improving Classification of Mature microRNA by Solving Class Imbalance Problem, Sci Rep, 6, 25941.
25. Zadrozny B., J. Langford, N. Abe (2003). Cost-sensitive Learning by Cost-proportionate Example Weighting, IEEE International Conference on Data Mining, 435-442.
26. Zou Q., M. Z. Guo, Y. Liu, J. Wang (2010). Unbalanced Classification Method and Its Application in Bioinformatics, Jisuanji Yanjiuyu Fazhan/Computer Research and Development, 47, 1407-1414 (in Chinese).

Assoc. Prof. Xia Geng
E-mail: gengxia@ujs.edu.cn

Xia Geng is an Associate Professor in School of Computer Science and Communication Engineering, Jiangsu University, Zhenjiang, Jiangsu, China. Her research interests include data mining, pattern recognition, and bioinformatics.

Prof. Yu-Quan Zhu
E-mail: yuquanzhu@ujs.edu.cn

Yu-Quan Zhu is a Professor and doctoral supervisor in School of Computer Science and Communication Engineering, Jiangsu University, Zhenjiang, Jiangsu, China. His research interests include data mining, pattern recognition, and bioinformatics.

Zhi Yang
E-mail: yangzhi@ujs.edu.cn

Zhi Yang is a lecturer in School of Management, Jiangsu University, Zhenjiang, Jiangsu, China. His research interests include data mining, pattern recognition, and information management.