GMR Genetics and Molecular Research
Online Journal
ISSN:1676-5680

imDC: an ensemble learning method for imbalanced classification with miRNA data

C.Y. Wang[1], L.L. Hu[2], M.Z. Guo[1], X.Y. Liu[1] and Q. Zou[2]

[1]School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China
[2]School of Information Science and Technology, Xiamen University, Xiamen, China

Corresponding authors: C.Y. Wang / M.Z. Guo
E-mail: chunyu@hit.edu.cn / maozuguo@hit.edu.cn

Genet. Mol. Res. 14 (1): 123-133 (2015)
Received March 14, 2014
Accepted October 20, 2014
Published January 15, 2015
DOI http://dx.doi.org/10.4238/2015.January.15.15

ABSTRACT. Imbalances typically exist in bioinformatics and are also common in other areas. A drawback of traditional machine learning methods is the relatively little attention given to small sample classification. Thus, we developed imDC, which uses an ensemble learning concept in combination with weights and sample misclassification information to effectively classify imbalanced data. Our method showed better results when compared to other algorithms with UCI machine learning datasets and microRNA data.

Key words: Bioinformatics; Ensemble learning; Imbalances; Machine learning; miRNA

INTRODUCTION

Classification has been a research hotspot in the field of machine learning in recent years, and several well-known methods for categorization were found to show good performance. In practice, however, differences in class imbalances have been reported to hinder the performance of various standard classifiers. Although traditional classification methods can be used to achieve better performance in balanced data, some small-category samples will be predicted incorrectly in order to achieve high overall classification accuracy. These samples play an important role in practical application. For example, erroneous diagnosis of patients may exert a psychological burden, while misdiagnosis will prevent administration of proper treatment in a timely manner. In addition, limitations due to unbalanced data exist in many other areas, such as oil exploration (Kubat et al., 1998), bank lending, medical diagnosis, information retrieval, and text classification. Improving the identification accuracy of a minority class in an unbalanced dataset has been thoroughly examined.

The imbalance phenomenon is more obvious for microRNA (miRNA). Unlike positive examples, which require validation in biological experiments, negative examples are generally identified in gene coding sequences at random. This leads to overrepresentation of a large number of negative samples and a small number of positive samples. In general, negative samples can be filtered out based on pre-miRNA hairpin secondary structures to control for the quantity and quality, but the filtering step does not always reduce the imbalance. Studies should be conducted to develop methods of effectively using positive and negative resources in bioinformatics.

A number of algorithms based on traditional machine-learning algorithms have been developed to improve the classification performance of imbalanced data sets. Research on classification methods for unbalanced are currently considered at two levels.

At the data level, imbalance can be eliminated or reduced by changing the data distribution. Most algorithms can be used to resolve data using 2 approaches: over-sampling and under-sampling. The first method increases minority class samples to improve classification performance of the minority class. The easiest method is to simply copy the minority class sample. This method leads to the natural introduction of additional training data and increases the training time, but does not add useful information to the sample, eventually leading to over-fitting. However, the synthetic minority over-sampling technique (SMOTE) (Chawla et al., 2002), a machine-learning approach based on over-sampling theory, could be used to avoid over-fitting but may introduce noise. Another method of reducing imbalances is by reducing the size of the majority class. This can be accomplished by randomly removing some of the samples in the majority class, which can lead to the loss of useful information. Drown et al. (2007) proposed an algorithm for identifying and removing noise samples to sample the majority class based on genetic algorithms.

Imbalance can also be corrected at the algorithm level to modify an existing classification algorithm. A cascaded algorithm for gradually reducing the number of samples in balanced data sets was proposed by Liu et al. (2006), in which a series of classifiers were introduced in training predict samples through an ensemble approach. An alternative method to the integrated approach is based on cost-sensitive strategies and cost information that can be acquired from a domain expert. There are several methods for utilizing cost-sensitive information such as the consideration-sensitive support vector machine proposed by Lee et al. (2004).

These methods use different approaches for improvement and optimization; however, classification accuracy remains low for the minority class. Unfortunately, when ensemble approaches are used, computation time quickly increases, making the use of this method impractical for large-scale data. The cost-sensitive learning method has been shown to be equivalent to sampling methods. However, random subsampling is universally used for data balance even though most useful information will be lost. Studies are necessary to improve unbalanced data sets.

Imbalance classification basis

Ensemble learning theory

In practice, the machine learning method is widely used in production, research, and daily life. Its numerous applications include science, speech recognition, face recognition, handwriting recognition, data mining, medical diagnosis, and games. Ensemble learning is a very important and popular branch of machine learning that applies the philosophy of '4 eyes see more than 2'; it learns a model with a range of learners, using specific rules to integrate various learning outcomes. Ultimately, ensemble learning yields more efficient machine learning compared to that possible by a single learner.

Traditional machine-learning methods attempt to identify a classifier closest to the actual classification function from a space formed by the various possible functions. Single classifier models include decision trees, neural networks, and naive Bayes classifiers, among others. Ensemble learning classifies new instances with integrated classifiers, combining multiple classification results in order to achieve better performance than possible using a single classifier. Figure 1 shows the basic concept behind ensemble learning.

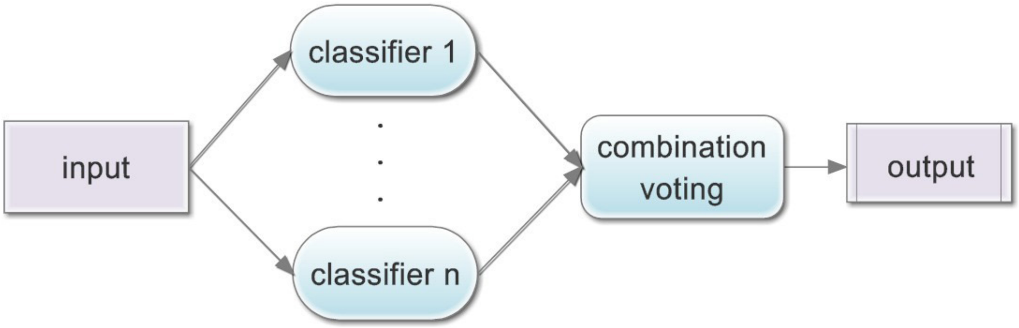

Figure 1. Ensemble classification ideology.

Schapire (1990) examined the original Boosting model and found that certain combinations of weak classifiers or base classifiers have the same properties as strong classifiers, indicating that identifying a strong classifier is unnecessary. Furthermore, weak classifiers require the specific classification error rate to be less than 0.5. Dietterich (2000, 2002) provided several reasons as to why ensemble learning is effective. First, statistically, because of the insufficient number of the training set instances, a learning algorithm cannot learn to target assumptions precisely, and there is some risk in allowing a learning algorithm to select a hypothesis. Therefore, offsetting the error between each assumption and goal assumption through integration is an advantage. Second, computationally, analysis of artificial neural networks and decision trees has revealed that the best learning hypothesis is a non-deterministic polynomial-time hard problem, as it is included in the other classifiers. We can only reduce the complexity to find the goal hypothesis through heuristic methods; furthermore, the assumption is not optimal. This indicates that integration of a number of assumptions will allow the end result to be closer to actual target function value.

As has been shown previously, an efficient solution for classification is to include experts in various specific areas, in which base classifiers are fully trained in the training set feature space and a combination of experts are necessary. The final efficient solutions will be related to not only the combination of base classifiers, but also the performance of base-classifier algorithms. Several excellent algorithms based on integrated learning, such as bagging boosting, play an important role.

Imbalance classification

Unbalanced data set problems will arise when particular sample types are overrepresented or underrepresented compared to others. Identifying minority classes with higher accuracy is necessary. However, there is no universally accepted definition regarding the size of the difference necessary to consider a dataset as unbalanced; in general, imbalance data problems appear when the data set shows a significant multiplied gap.

The University of California Irvine (UCI) machine-leaning database contains 187 datasets, with the number continually increasing. With the emergence of increasingly large numbers of unbalanced datasets, the UCI is often used for standard test datasets as it contains a wide variety of data, including national diabetes research, computer virus, and Trojan information.

A traditional machine-learning algorithm aims to improve the overall recognition rate but will sacrifice recognition accuracy for a minority class. For example, for a data distribution such as 100:900, 90% classification accuracy is reached even when dividing all samples into the majority class. A further challenge of machine learning is to identify an accurate model for unbalanced datasets. Various unbalanced classification algorithms have been developed. SMOTE, one-sided selection, improved SVM algorithm, cost-sensitive algorithms, and ensemble classification algorithms all form an important functional layer for unbalanced classification.

In machine learning, a series of indicators are necessary to evaluate classifier performance. The indicators are generally defined as follows: *tp* indicates the number of true-positive predictions, *tn* indicates the number of true-negative predictions, *fp* indicates the number of false-positive predictions, and *fn* indicates the number of false-negative predictions. Evaluation criteria of classification performance are as follows:

1) Sensitivity (*se*) reflects the classification accuracy of positive examples based on the following formula:

$$se = \frac{tp}{tp + fn} \qquad \text{(Equation 1)}$$

2) Specificity (*sp*) reflects classification accuracy of negative examples based on the following formula:

$$sp = \frac{tn}{fp + tn} \qquad \text{(Equation 2)}$$

3) Accuracy (*acc*) is a commonly used indicator in classification problems as it reflects the overall classification performance of the classifier based on the following formula:

$$acc = \frac{tn + tp}{tp + tn + fp + fn} \qquad \text{(Equation 3)}$$

4) Precision reflects the proportion of true predictions to all predictions of a category:

$$precision = \frac{tp}{tp + fp} \qquad \text{(Equation 4)}$$

5) Recall reflects the fraction of all samples that are predicted to be true:

$$recall = \frac{tp}{tp + fn} \qquad \text{(Equation 5)}$$

6) The F_{measure} value is the comprehensive classification performance response of recall and precision:

$$F_{\text{measure}} = \frac{(1+\sigma^2)\text{*precision+recall}}{\sigma^2\text{precision+recall}} \qquad \text{(Equation 6)}$$

In equation (6), σ ≤ [0, ∞]; when σ = 0, the F value is equal to precision. When σ = ∞, the F value is equal to recall. In general, σI s set to 1 and the F value is the harmonic mean of recall and precision.

7) The Matthew correlation coefficient (MCC) uses the number of correct and incorrect predictions to measure classifier performance:

$$\text{MCC} = \frac{(\text{tp*tn})\text{-}(\text{fp*fn})}{\sqrt{(tp + fn)*(tp + fp)*(tn + fp)*(tn + fn)}} \qquad \text{(Equation 7)}$$

8) A receiver operating characteristic (ROC) curve is an excellent robustness indicator of classifier performance; a value closer to the upper left corner is better. An ROC curve is a visual, but not quantitative, evaluation of classifier performance, for which the area under the ROC curve can be used to quantitatively evaluate the area under the curve, as shown in Figure 2.

Commonly used indicators for unbalanced datasets include precision, recall, and F value, in which precision and recall of the minority class can be used to effectively evaluate the classification performance of the minority class in unbalanced data. Precision and recall F value are also useful as evaluation criteria.

Acc is generally used to assess overall classification performance, but is ineffective for unbalanced datasets. *Acc* is approximately equivalent to *sp*, as *tn* and *fp* are much larger than *tp* and *fn* in the case of unbalanced data, while *se* is equivalent to recall in pattern recognition. Therefore *se* and *sp* are used as the standards for measuring the effect of unbalanced classification in bioinformatics.

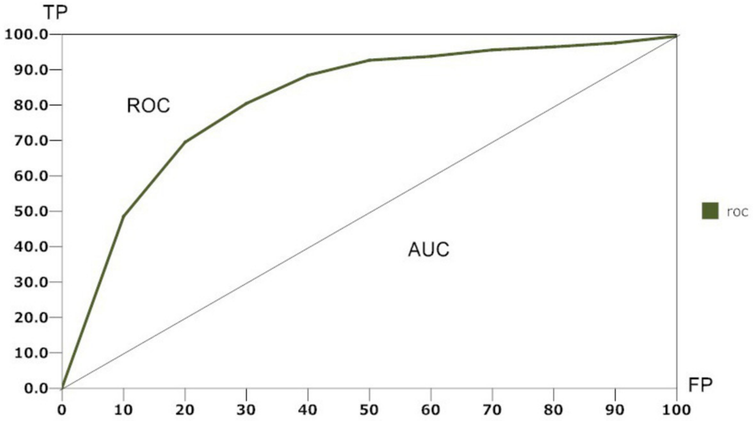

Figure 2. Receiver operating characteristic (ROC) curve diagram.

MATERIAL AND METHODS

Because misclassification of the minority class will result in very large losses in life and work, it would be reasonable to adopt unbalanced data classification algorithms in such cases; however, these algorithms may not be effective for resolving issues in unbalanced datasets with different distributions. Ensemble learning can be applied to such imbalanced datasets (imDC).

A study by Krogh (1995) examining ensemble classification found that a large difference between base classifiers should improve the effect of integration. In addition, some classification algorithms appear to incorporate learning, but are unable to train for other datasets. Thus, 16 classification algorithms, including decision trees, random forests, support vector machines, naive Bayes, and k neighbors, among others, are selected, and identifying 5 excellent classification algorithms is necessary for circular training of datasets. Note that a new dataset that contains minority class examples and an equal number of random majority class examples is used to evaluate 16 algorithms to produce 5 optimal algorithms. Most classifiers are weak and use default parameters with no tuning. Because excessive classifications may affect the final ensemble result, 5 optimal classification algorithms are appropriate.

The number of base classifiers (iterationNum) depends on the ratio of majority class examples and minority class examples and is referred to as n. If $n \leq 5$, we set iterationNum to 5, otherwise iterationNum is n. When the ratio was small, the 5 optimal classification algorithms were adequately trained and the solution performed optimally after integration. In addition, useful information for majority class examples was retained when the ratio was large. Next, training data were trained iterationNum times and a classification algorithm chosen from the 5 optimal classification algorithms was used for training the dataset each time.

Some samples were misclassified because of the unbalanced distribution of data or noise; importantly, these samples may have contained very valuable information, referred to as the richest information. In the training process, the weights of the sample containing the richest information began increasing, making it easier to select these samples for the next round of base classifiers. Because minority class samples and majority class samples randomly selected according to their weights formed a new sub-dataset to train our classification algorithm, we set the weight to the majority class to ensure selection of majority class samples with the richest information in order of priority.

We used geometric evaluation of the misclassification rate of positive and negative samples as the weight of base classifiers in each round before integrated classification. A flag may be used as a first step, as shown in Table 1.

Table 1. Sample tag.

	Predicted class	
	True	False
True class	0	1
True	2	3
False		

The flags were designated as follows. If a sample belonged to the minority class and prediction was correct, then flag = 0; if a sample belonged to the minority class and prediction was incorrect, then flag = 1; if a sample belonged to the majority class and the prediction was correct, then flag = 3; if a sample belonged to the majority class and the prediction was incorrect, then flag = 4. The parameter g was related to the weight of base classifiers, assuming the 4 types of samples were n0, n1, n3, and n2 after the test, and g = [1.0 x n0/(n0 + n1) + 1.0 x n3/(n3 + n2)]/2.

The basic flow of the algorithm was as follows:

Algorithm 1

Input: Data set D; category of minority class sample

Output: Predicted category of test sample x

ImDC:

Step 1: Begin;

Step 2: Obtained a new dataset D', which contained minority class examples and an equal number of random majority class examples (number of minority class examples, majority class examples, and D' are n_+, n_-, $2n_+$);

Step 3: 5 optimal classification algorithms were selected by 5-fold cross validation with the dataset D' from 16 classification algorithms, number 1-5;

Step 4: Minority class samples: D_+, majority class samples: D_- Samples in D_- were assigned to initial weights: $w_i = 1/ n_-$, $i = 1, 2… n_-$;

Step 5: If $n_+/n_- \leq 5$ iterationNum = 5; otherwise iterationNum = n_+/n_-;

Step 6: For j = 1 to iterationNum;

Step 7: D_+ and $2n_+$ samples extracted from D_- (if insufficient, select all) formed a new data set: D_{train};

Step 8: Selected the j% 5 numbered classification algorithm numbered j% 5 to train T_j and generate base classifiers h_j;

Step 9: testing D_-, D_+ with T_j, and recording the number of positive and negative misclassification samples misclassification: n_{+_w}, n_{-_w}. Modified the weights of minority class samples: $w = w+(n_{+\_w})/ D_-$ followed by normalization. Setting the weight of current round of base classifier: $g_j = [1.0*(D_- - n_{+\_w})/ D_- + 1.0*(D_+ - n_{-\_w}) / D_+]/2$;

Step 10: End for;

Step 11: Weighted voting for predictions with iterationNum integrated classifiers (h_1 ~$h_{iterationNum}$):

$H(x) = \sum_{j=1}^{iterationNum} g_j h_j(x)$

Step 12: End

RESULTS AND DISCUSSION

Our experiment was based on UCI test data and miRNA bioinformatics data. We used these data to verify the effectiveness of our method proposed and strategy. Experiments were conducted in 2 groups: UCI test and miRNA test.

UCI data

The five representative datasets in the UCI included cmc, haberman, ionosphere, letter, and pima, as shown in Table 2. The samples of these 5 datasets were set to be unbalanced so that the smallest category was regarded as the positive class and the rest were treated as the negative class. These datasets were quite different in scale, as the total number ranged from 306-20,000 and the proportion of the minority class and majority class ranged from 2-25. As described above, these datasets were derived from the UCI database and covered typical classification problems in machine learning and pattern recognition. Therefore, they are strongly represented imbalanced datasets.

Table 2. 5 University of California Irvine data.

Dataset	Size	Class	Positive class	|P|:|N|
Cmc	1473	3	Class = 2	1140:333
Haberman	306	2	Class = 2	225:81
Ionosphere	351	2	Class = good	225:126
Letter-recognition	20000	26	Class = A	19211:789
Pima-indians-diabetes	768	2	Class = 1	500:268

To quantitatively evaluate our algorithm, we selected AdaBoost, UnderSampl, HSampl, AsymBoost, BalanceCascade, and LibID as comparative methods. AdaBoost is used in many applications because of its stable performance; each round of already established base classifiers affects the subsequent round of base classifier performance. The UnderSampl sampling method is a common random under-sampling method that chooses positive examples and the same ratio of randomly negative examples as the new balanced training set. After data processing using the UnderSampl method, the AdaBoost algorithm was used to train classifiers. For HSampl, we used over-sampling to double the small samples, and then used under-sampling to shrink the large samples. Finally, AdaBoost was used to train classifiers when the dataset was balanced. Since AdaBoost treats misclassified positive and negative examples equally, it cannot support high performance in imbalanced data. Asymboost is an improved algorithm of AdaBoost in which positive samples had a higher price when they were misclassified (Lin et al., 2013, 2014; Song et al., 2014). However, when positive and negative examples had an equal cost for classification, the algorithm was equivalent to AdaBoost. BalanceCascade uses a cascade structure to gradually narrow the major categories and gradually create a more balanced dataset; thus, the algorithm uses a series of classifiers to complete the integrated classification. LibID is a solution to solve imbalance classification as proposed by Quan (2010), which aims to split major categories data and using a vote to obtain the result. Results are shown in Figure 3.

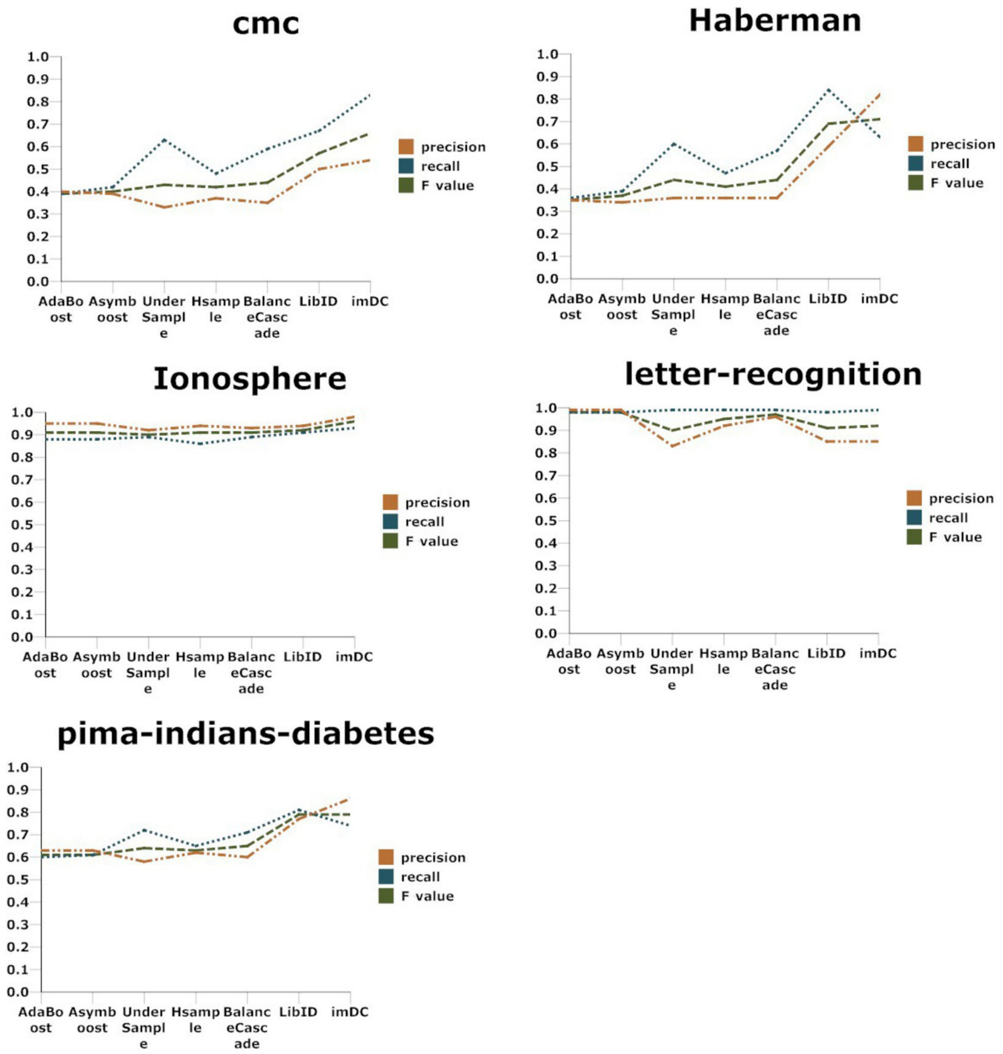

Figure 3. Cross-validation results of five UCI data sets.

miRNA data

miRNA is an important non-coding RNA molecule that plays an important role in regulating gene expression. Specifically identifying sequences in a biological genome is an important use of miRNA. Identified fragments can be analyzed using biological methods such as biochips for authentication, which is costly and complex. Because miRNAs need to be examined experimentally for their accuracy, the quantity of miRNA needed is small. However, miRNAs must form stable hairpins for processing, and a large number of sequences similar to hairpin precursor miRNAs exist in millions of genes. Positive examples and negative examples show important differences, and typical imbalance exists. Xue et al. (2005) presented the triplet-SVM approach, which is capable of solving unbalanced classification. They found datasets of miRNAs with 193 positive examples and 8494 negative examples, in which 30 positive examples and 1000 negative examples were a test set. In order to evaluate the imDC method, we compared triplet-SVM and libID using the same dataset containing a test dataset of 30 positive examples, 1000 negative examples, and a training dataset of the remaining 163 positive examples and 7494 negative examples. The results are shown in Table 3.

Table 3. Cross-validation results of 5 UCI data sets.

	Sn	Sp
Triplet-SVM	0.93	0.88
LibID	0.83	0.92
ImDC	0.86	0.93

Sn = sensitivity; Sp = specificity.

CONCLUSIONS

miRNA is an important non-coding RNA that is important in the development of various diseases. An imbalance of positive and negative miRNA examples is observed when using machine-learning methods to resolve issues related to miRNA identification; therefore, we propose an effective method referred to as imDC. imDC makes full use of minority class samples, increases misclassified sample weights, and can effectively handle imbalanced data using a special distribution. Experiments on multiple sets of data in the UCI and miRNA databases showed satisfactory results.

Although imDC can be used to effectively predict the classification of minority class samples, time performance of our 5 optimal classifiers was not considered, and some classifiers showed a large increase in computation time. In addition, integrated classifiers use default parameters without tuning, which requires further investigation.

ACKNOWLEDGMENTS

Q. Zou is supported by the Natural Science Foundation of China (#61370010). M.Z. Guo is supported by the Natural Science Foundation of China (#61271346) and the Specialized Research Fund for the Doctoral Program of Higher Education of China (#20112302110040). X.Y. Liu is supported by the Natural Science Foundation of China (#61172098 and #91335112).

REFERENCES

Chawla NV, Bowyer KW, Hall LO and Kegelmeyer WP. (2002). SMOTE: Synthetic Minority Over-Sampling Technique. Journal of Artificial Intelligence Research. 16:321-357

Diettrich TG (2000). Ensemble methods in machine learning. MCS ’00 Proceedings of the First International Worshop on Multiple Classifier Systems. Cagliari, Italy.

Dietterich TG (2002). Ensemble learning. In: The Handbook of Brain Theory and Neural Networks, 2nd edn. *The MIT Press, Cambridge.*

Drown DJ, Khoshgoftaar TM and Narayanan R (2007). Using evolutionary sampling to mine imbalanced data. The 6th International Conference on Machine Learning and Applications. Washington DC: IEEE Computer Society 363-368.

Krogh A, et al. (1995). Neural network ensembles, cross validation, and active learning. In: Advances in neural information processing systems. MIT Press, Cambridge, 231-238.

Kubat M, Holte RC and Matwin S (1998). Machine learning for the detection of oil spills in satellite radar images. *Machine Learning* 30: 195-215.

Lee Y, Lin Y and Wahba G (2004). Multicategory support vector machines: theory and application to the classification of microarray data and satellite radiance data. *Journal of the American Statistical Association* 99: 67-81.

Lin C, Zou Y, Qin J, Liu X, et al. (2013). Hierarchical classification of protein folds using a novel ensemble classifier. *PLoS One* 8: e56499.

Lin C, Chen W, Qiu C, Wu Y, et al. (2014). LibD3C: Ensemble classifiers with a clustering and dynamic selection strategy. *Neurocomputing* 123: 424-435.

Liu XY, Wu JX and Zhou ZH (2006). A cascade-based classification method for class-imbalanced data. *J. Nanjing Univ. (Natural Sci.)*. 42: 148-155.

Schapire RE (1990). The strength of weak learnability. *Machine Learning* 5: 197-227.

Song L, Li D, Zeng X, Wu Y, et al. (2014). nDNA-prot: identification of DNA-binding proteins based on unbalanced classification. *BMC Bioinformatics* 15: 298.

Quan Z, Maozu G, Yang L and Jun W (2010). A classification method for class-imbalanced data and its application on bioinformatics. *J. Computer Research and Development* 47:1407-1414.

Xue CG, Fei L, Tao H, Liu GP, et al. (2005). Classification of real and pseudo microRNA precursors using local structure-sequence features and support vector machine. *BMC Bioinformatics*, 6: 310