Bio-Medical Materials and Engineering 26 (2015) S1855–S1862
DOI 10.3233/BME-151488
IOS Press

Classification of imbalanced bioinformatics data by using boundary movement-based ELM

Ke Cheng[a], Qingfang Chen[b], Xibei Yang[a], Shang Gao[a] and Hualong Yu[a,*]
[a]*School of Computer Science and Engineering, Jiangsu University of Science and Technology, No. 2 Mengxi Road, Zhenjiang 212003, China*
[b]*School of Electronic Information, Jiangsu University of Science and Technology, No. 2 Mengxi Road, Zhenjiang 212003, China*

Abstract. To address the imbalanced classification problem emerging in Bioinformatics, a boundary movement-based extreme learning machine (ELM) algorithm called BM-ELM was proposed. BM-ELM tries to firstly explore the prior information about data distribution by condensing all training instances into the one-dimensional feature space corresponding to the original output in ELM, and then on the transformed space, to find the optimal moving distance of the classification hyperplane by estimating the probability density distributions of the instances in different classes. Experimental results on four real imbalanced bioinformatics classification data sets indicated that the proposed BM-ELM algorithm outperforms some traditional bias correction algorithms due to it can greatly improve the sensitivity of the classification results with small loss of specificity as possible. Also, BM-ELM algorithm has presented better performance than the widely used support vector machine (SVM) classifier. The algorithm can be widely popularized in various large-scale bioinformatics applications.

Keywords: Bioinformatics, extreme learning machine, imbalanced classification, kernel density estimation

1. Introduction

In post-genome era, with the widely use of various high-throughput technologies, biological data has sharply increased [1, 2]. To discovery significant information to understand complex biological processes and develop meaningful applications from large-scale biological data, machine learning technologies can be applied [3]. Extreme learning machine (ELM), which is proposed by G.B. Huang, et al. [4, 5], has shown the specific superiority in bioinformatics [6], as it has fast response for those large-scale classification tasks, as well has stronger generalization capability than those traditional classification algorithms [7]. Unfortunately, like other classifiers, ELM may also learn undesirable class boundaries from data with skewed class distributions [8].

*Address for Correspondence: Hualong Yu, School of Computer Science and Engineering, Jiangsu University of Science and Technology, No. 2 Mengxi Road, Zhenjiang 212003, China. Tel.: +86 511 88889594; Fax: +86 511 84409018; E-mail: yuhualong@just.edu.cn.

To deal with class imbalance problem in the context of ELM, previous work has considered to adopt several traditional bias correction strategies. Zong, et al. [8] profited from the idea of cost-sensitive learning to present a weighted ELM classifier (WELM). By designating different penalty factors for the training errors belonging to different categories, the performance of the minority classes can be highlighted. Vong, et al. [9] adopted a modified random oversampling method named prior duplication to promote the recognition rate of the level of suspended particulate matter. Sun, et al. [10] integrated synthetic minority oversampling technology (SMOTE) [11] into a multiple ELMs framework to improve the prediction of corporate life cycle.

This article tries to present a novel and simple class imbalance learning algorithm named BM-ELM that corrects the bias by moving the classification hyperplane towards the region of the majority class. BM-ELM first trains a basic ELM classifier using training instances, then projects all instances from the original feature space to a one-dimensional decision space (corresponding to the one-dimensional actual output in ELM). Next, kernel density estimation (KDE) [12] technology is adopted to estimate the probability density distributions of two different classes on the condensed one-dimensional space. Finally, the horizontal axis location corresponding to the intersecting point of the two probability density curves is found to calculate the movement direction and distance of the classification hyperplane.

BM-ELM algorithm was compared with several traditional bias correction strategies on four real imbalanced bioinformatics data sets [13-15]. The results indicate that BM-ELM can greatly improve the sensitivity with small loss of specificity as possible. Therefore, BM-ELM algorithm is a competitive candidate to deal with large-scale skewed classification tasks in bioinformatics.

2. Methods

2.1. Extreme learning machine

Extreme learning machine (ELM) is a fast learning algorithm to train single hidden layer forward networks (SLFNs). Unlike those traditional learning algorithms, e.g., back-propagation (BP) algorithm, ELM randomly generates the weights and bias between the input layer and the hidden layer, then adopts the least-square algorithm to acquire the solution of the hidden layer output weights. Suppose the hidden layer output (with L nodes) can be presented by a row vector $\mathrm{h}(\mathrm{x}) = [\mathrm{h}_1(\mathrm{x}), \ldots, \mathrm{h}_L(\mathrm{x})]$, where x is the input instance. Given N training instances $(\mathrm{x}_i, \mathrm{t}_i)$, the mathematical model of the SLFNs is:

$$\mathrm{H}\beta = \mathrm{T} \tag{1}$$

where $\mathrm{H} = \begin{bmatrix} \mathrm{h}(\mathrm{x}_1) \\ \ldots \\ \mathrm{h}(\mathrm{x}_N) \end{bmatrix}$ is the hidden layer output matrix, β is the output weight matrix and T is the target vector. Then the least square solution with minimal norm is determined using Moore-Penrose "generalized" inverse $\widehat{\mathrm{H}}$:

$$\beta = \widehat{\mathrm{H}}^{\dagger}\mathrm{T} = \begin{cases} \mathrm{H}^T(\frac{\mathrm{I}}{C} + \mathrm{H}\mathrm{H}^T)^{-1}\mathrm{T}, \text{when } N \le L \\ (\frac{\mathrm{I}}{C} + \mathrm{H}^T\mathrm{H})^{-1}\mathrm{H}^T\mathrm{T}, \text{when } N > L \end{cases} \tag{2}$$

As can be seen from the two equations above, a positive identity matrix I/*C* is added to the diagonal of HH^T or H^TH for acquiring a better generalization performance [7].

The same solution can also be obtained by using the optimization method that is described as follows:

$$\begin{aligned} &Lp_{ELM} = \frac{1}{2}||\beta||^2 + \frac{1}{2}C\sum_{i=1}^{N}||\xi_i||^2 \\ &\text{Subject to: } \mathrm{h}(\mathrm{x}_i)\beta = t_i^T - \xi_i^T, i = 1,\ldots,N \end{aligned} \tag{3}$$

where $\xi_i = \left[\xi_{i,1}, \ldots, \xi_{i,m}\right]^T$ is the training error vector of the m^{th} output nodes with respect to the training instance x_i, and *C* is the regularization parameter to present the trade-off between the training errors and the generalization ability. The optimization equation in Eq. (3) can be solved by Karush-Kuhn-Tucker (KKT) theorem [16], and the solution is the same as presented in Eq. (2).

2.2. One-dimensional projection and kernel density estimation

It is well-known that in ELM, the class label of an instance is finally determined by the actual output. When the actual output is larger than 0, the instance should be divided into the positive class, otherwise it has to be put into the negative class. Generally speaking, for imbalanced classification problem, the minority class and the majority class are denoted as the positive and negative classes, respectively. Therefore, after constructing an ELM classification model, all training instances can be projected from the original feature space to a one-dimensional decision space, where the coordinate of each instance is denoted by its actual output.

Next, kernel density estimation (KDE) [12], which is a non-parametric way to estimate the probability density function of a random variable, is adopted to estimate the probability density distributions of the instances belonging to the two different classes on the projected one-dimensional space, respectively. Specifically, Gaussian kernel is adopted as the kernel function in KDE. Then as discussed in [17], the probability density distribution approximately equals to the distribution of the posterior probability, thus the optimal break point between two classes should be the intersecting point of the two density curves. The distance between the intersecting point and the original point denotes the optimal movement distance of the original classification hyperplane.

2.3. Description of the BM-ELM algorithm

According to what is described in Section 2.2, BM-ELM algorithm can be divided into three stages. At the first stage, a baseline ELM classifier is constructed on the original training set. At the second stage, all instances are projected on a one-dimensional space according to the distance between each example and the initial classification hyperplane, then the probability density distributions of the two different classes are estimated on the projected one-dimensional space. Finally, at the third stage, the intersecting horizontal axis location of the probability density distribution curves belonging to the two different classes is recorded as the movement distance of the classification hyperplane. The detailed description of BM-ELM algorithm is provided as follows:

Algorithm: BM-ELM
Input: Training set *Tr*, test set *Te*, the number of hidden nodes *L* and the regularization parameter *C*.
Output: The threshold of the movement distance *thr*, the classification results on *Te*.
Procedure:

(1) Adopt *Tr* to train a basic ELM classifier *Learner* which has *L* hidden nodes, uses *C* as regularization parameter and randomly generates the hidden layer's weights and bias in the region of (-1,1);
(2) Obtain all actual outputs corresponding to the instances in *Tr*, and then project them into a one-dimensional space;
(3) Use kernel density estimation (KDE) approach to obtain the probability density distribution curves of two different classes, respectively;
(4) Find the intersecting horizontal axis location of the two density curves and record it as *x*, then calculate the threshold of the movement distance as *thr*=0-*x*;
(5) Obtain all actual outputs of the instances in *Te* by *Learner* acquired at step 1, and then for each actual output, add the threshold *thr* acquired at step 4, finally provide the classification results on the test set *Te*.

Figure 1 gives the graphical representation to describe the procedure of BM-ELM algorithm on two synthetic data sets. The data sets both have 1100 synthetic instances with 10:1 class imbalance ratio. The difference of them is the former has larger class margin. From Figure 1, it can be observed that BM-ELM algorithm is able to automatically find the optimal movement distance and move the classification hyperplane to the most appropriate position.

3. Experiments

3.1. Data sets and experimental settings

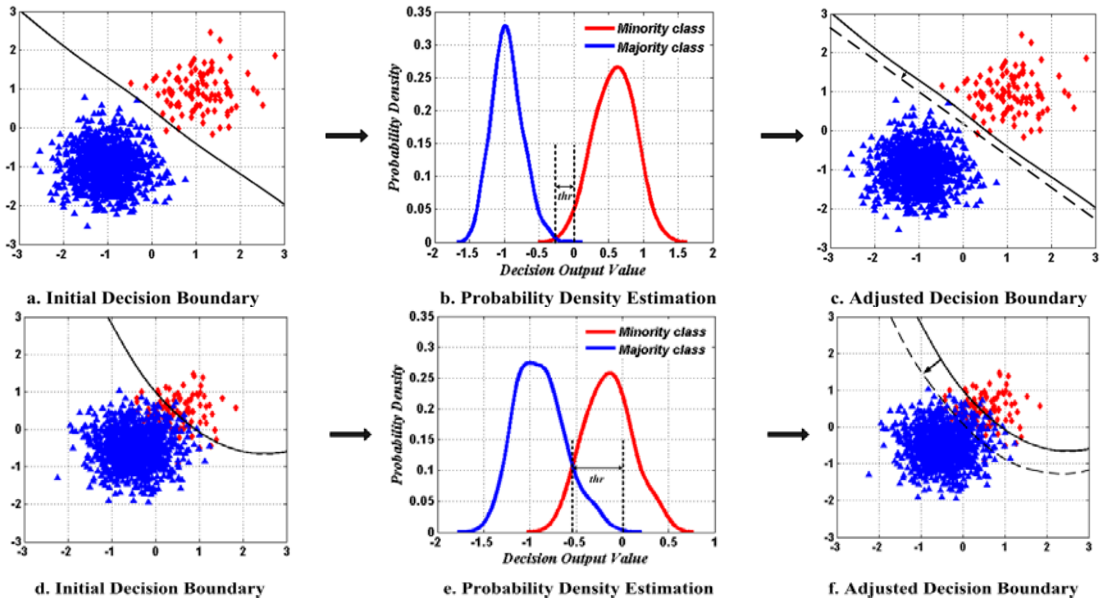

Fig. 1. Procedure of BM-ELM algorithm on two synthetic data sets (the majority class, i.e., the negative class, has 1000 instances; the minority class, i.e., the positive class, has 100 instances) that satisfying different Gaussian distributions (on data set 1: μ_+=1, μ_-=-1, σ=0.5; on data set 2: μ_+=0.5, μ_-=-0.5, σ=0.5). The subgraphs on the two rows denote the initial construction of the decision boundary with *L*=10 and *C*=10 in ELM, Kernel density estimation and the determination of the final decision boundary on the two data sets, respectively. It is clear that BM-ELM can explore the prior data distribution and automatically provide the optimal boundary movement distance.

Table 1
Four imbalanced bioinformatics data sets used in this paper

Data set	Number of attributes	Number of instances	Class imbalance ratio
MicroRNA precursors	32	8687	44.01
SNP	25	3074	15.80
Box H/ACA snoRNA	14	8510	129.92
Box C/D snoRNA	14	45515	147.74

Four imbalanced bioinformatics data sets were used to evaluate the performance of the proposed BM-ELM algorithm, including MicroRNA precursors [13], SNP [14], Box H/ACA snoRNA and Box C/D snoRNA [15]. MicroRNA precursor data set is required to construct a classifier to distinguish 193 real microRNA precursors (positive instances) from 8687 candidates [13]. SNP data set includes 3074 SNP candidates acquired from Human EST sequences, and among them, there are only 183 real SNPs (positive instances) [14]. Box H/ACA snoRNA and Box C/D snoRNA data sets are constructed to distinguish Box H/ACA snoRNA and Box C/D snoRNA from lots of non-coding RNA (ncRNA) instances. Box H/ACA snoRNA contains 65 positive instances and 8445 negative ones, while Box C/D snoRNA contains 306 positive instances and 45209 negative ones. The detailed descriptions about these data sets are provided in Table 1.

To present the superiority of the proposed BM-ELM algorithm, it has been compared to the basic ELM classifier and several traditional bias correction algorithms constructed in the context of ELM, including the weighted ELM (WELM) [8], ELM with random undersampling (ELM-RUS), ELM with random oversampling (ELM-ROS) and ELM with SMOTE (ELM-SMOTE) [10]. Moreover, SVM, as a widely used classifier in bioinformatics, has also been implemented to compare with the proposed algorithm. In particular, random undersampling was adopted to combine with SVM (SVM-RUS) for solving class imbalance problem.

In addition, to guarantee the impartiality of the comparative results, grid search was adopted to search the optimal parameters in ELM and SVM. For ELM, *sigmoid* function was used as activation function on the hidden level, and two other parameters L and C were selected from $\{10, 20, \ldots, 200\}$ and $\{2^{-20}, 2^{-18}, \ldots, 2^{20}\}$, respectively. For SVM, *rbf* kernel was used as the kernel function, and the two parameters C and σ were both extracted from $[2^{-8}, 2^{-7}, \ldots, 2^{8}]$. Specifically, SVM was implemented by libsvm matlab toolbox with the version 3.1.8 [18].

Finally, three performance evaluation metrics: sensitivity, specificity and g-mean, were adopted to compare various classification algorithms. Each experiment executed 10 times' five-fold cross validation, and then provided the average classification results.

3.2. Results and discussions

Table 2 gives the classification results of the seven classification algorithms on the four used bioinformatics data sets.

From Table 2, the following facts can be observed:

- All bias correction strategies are helpful to improve the performance of imbalanced classification data, as they always acquire higher sensitivity and g-mean values than that in the basic ELM classifier. Actually, g-mean reflects the trade-off between the accuracy of the positive class and the accuracy of the negative class, thus the higher g-mean is, the more balanced the accuracies between two classes are. ELM without considering bias correction often misclassifies majority or even all instances belonging to the minority class, causing the classification results meaningless.

Table 2

Classification results of the seven classification algorithms on the four used bioinformatics data sets

Data set	ELM	WELM	ELM -RUS	ELM -ROS	ELM -SMOTE	SVM -RUS	BM -ELM
Sensitivity							
MicroRNA precursors	0.0485	0.8600	0.9048	0.8940	0.8767	0.8845	0.8887
SNP	0.0133	0.6116	0.6468	0.6009	0.6047	0.6472	0.6120
Box H/ACA snoRNA	0.0000	0.9072	0.9571	0.9302	0.9341	0.9554	0.9588
Box C/D snoRNA	0.0000	0.9191	0.9210	0.9297	0.9331	0.9537	0.9619
Specificity							
MicroRNA precursors	1.0000	0.9005	0.8428	0.9120	0.9280	0.9058	0.9137
SNP	0.9981	0.7929	0.6500	0.7831	0.7896	0.7483	0.8107
Box H/ACA snoRNA	1.0000	0.9537	0.9027	0.9433	0.9490	0.9095	0.9529
Box C/D snoRNA	1.0000	0.9490	0.8936	0.9369	0.9477	0.9298	0.9518
G-mean							
MicroRNA precursors	0.2003	0.8793	0.8729	0.9026	0.9013	0.8951	0.9008
SNP	0.0671	0.6953	0.6469	0.6850	0.6900	0.6927	0.7034
Box H/ACA snoRNA	0.0000	0.9271	0.9287	0.9379	0.9407	0.9339	0.9554
Box C/D snoRNA	0.0000	0.9335	0.9069	0.9330	0.9401	0.9419	0.9571

- In contrast with four other strategies in the context of ELM, BM-ELM can effectively promote sensitivity, and at the same time guarantee small loss of specificity as possible. In other word, BM-ELM algorithm can improve g-mean metric to a large extent. This is a useful and significant result, as in bioinformatics applications, sensitivity and specificity are both important performance metrics, and they are expected to be increased simultaneously.
- Oversampling strategies, including ROS and SMOTE, often outperform undersampling and weighting strategies. It is not difficult to understand this phenomenon, because undersampling removes some important classification information, and weighting tends to pre-assign empirical but not optimal weights. However, oversampling always consumes excessive temporal and spatial costs, especially on those highly skewed data sets, e.g., Box C/D snoRNA data set.

BM-ELM performs better than widely used SVM-RUS algorithm, although in previous work, ELM has presented quite similar performance with SVM [4, 5, 7]. Actually, BM-ELM partially explores the prior information about the instance distributions, which can adaptively find the optimal position of the classification hyperplane. Here, optimal requires a particular condition, i.e., training set and test set have to share an identical distribution. Therefore, BM-ELM profits more from the embedded bias correction strategy than SVM-RUS. In addition, compared to SVM, ELM has significantly faster training speed, thus BM-ELM has specific superiority in large-scale imbalanced bioinformatics applications.

Figure 2 gives the average moving distance and the corresponding standard deviation of the classification hyperplane on each data set. From Figure 2, it is not difficult to observe that the moving distance depends on the class imbalance ratio of the data set more or less. On the two data sets whose class imbalance ratio are higher than 100, the optimal movement distances are obviously larger than that on two others. In fact, the moving distance not only depends on the class imbalance ratio, but also has close relationship with the size of class overlapping [17].The higher the class imbalance ratio is, the larger the class overlapping size is, then the longer the moving distance should be. Fortunately, BM-ELM can adaptively find the optimal moving distance by exploring the real prior data distribution.

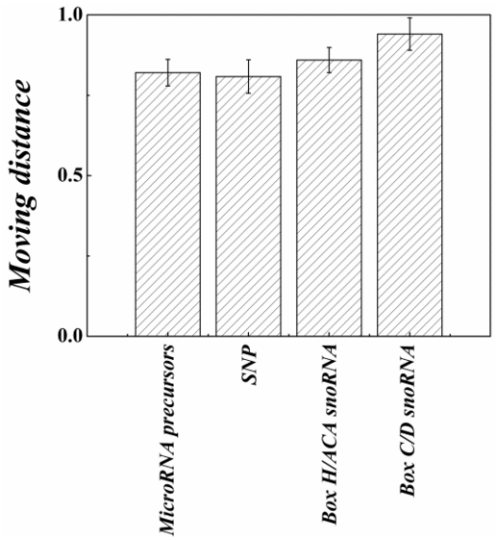

Fig. 2. In BM-ELM, the average moving distance and the corresponding standard deviation of the classification hyperplane on each data set.

4. Conclusion

In this paper, a novel algorithm called boundary movement-based extreme learning machine (BM-ELM) was proposed and applied to classify imbalanced bioinformatics data. Unlike those traditional bias correction strategies, BM-ELM explores partial prior information about the original data distribution by condensing instances into an one-dimensional decision space and estimating the probability density distributions of the two classes in the condensed space. Experimental results on four real imbalanced bioinformatics classification tasks showed that BM-ELM algorithm is effective and efficient to address class imbalance problem in large-scale bioinformatics applications. Moreover, adopting ELM as baseline classifier helps to save time consumption to a large extent.

In future work, new ELM-based class imbalance learning algorithms with coupling more prior distribution information are expected to be proposed. Also, BM-ELM algorithm is promising to be applied in more extensive bioinformatics applications. At last, the possibility of transforming BM-ELM algorithm to deal with multiclass imbalance problem existing in bioinformatics will be investigated in the future work, too.

Acknowledgments

This work was supported by National Natural Science Foundation of China under Grant No. 61305058, Natural Science Foundation of Jiangsu Province of China under Grant No. BK20130471, Open Project Program of Key Laboratory of Modern Agriculture Equipment and Technology (Jiangsu University), Ministry of Education, Jiangsu Province, China under Grant No. NZ201303, China Postdoctoral Science Foundation under Grant No. 2013M540404, Jiangsu Planned Projects for Postdoctoral Research Funds under Grant No. 1401037B, and Qing Lan Project of Jiangsu Province of China.

References

[1] U. Yu, S.H. Lee, Y.J. Kim and S. Kim, Bioinformatics in the post-genome era, Journal of Biochemistry and Molecular Biology **37** (2004), 75–82.
[2] M. Kanehisa and P. Bork, Bioinformatics in the post-sequence era, Nature Genetics **33** (2003), 305–310.
[3] P. Larranaga, B. Calvo, R. Santana, C. Bielza, J. Galdiano, I. Inza, J.A. Lozano, R. Armananzas, G. Santafe, A. Perez and V. Robles, Machine learning in bioinformatics, Briefings in Bioinformatics **7** (2003), 86–112.
[4] G.B. Huang, Q.Y. Zhu and C.K. Siew, Extreme learning machine: Theory and applications, Neurocomputing **70** (2006), 489–501.
[5] G. Huang, G.B. Huang, S. Song and K. You, Trends in extreme learning machine: A review, Neural Networks **61** (2015), 32–48.
[6] R. Zhang, G.B. Huang, N. Sundararajan and P. Saratchandran, Multicategory classification using an extreme learning machine for microarray gene expression cancer diagnosis, IEEE/ACM Transactions on Computational Biology and Bioinformatics **4** (2007), 485–495.
[7] G.B. Huang, H. Zhou, X. Ding and R. Zhang, Extreme learning machine for regression and multiclass classification, IEEE Transactions on System, Man and Cybernetics: Part B: Cybernetics **42** (2012), 513–529.
[8] W. Zong, G.B. Huang and Y. Chen, Weighted extreme learning machine for imbalance learning, Neurocomputing **101** (2013), 229–242.
[9] C.M. Vong, W.F. Ip, P.K. Wong and C.C. Chiu, Predicting minority class for suspended particulate matters level by extreme learning machine, Neurocomputing **128** (2014), 136–144.
[10] S.J. Sun, C. Chang and M.F. Hsu, Multiple extreme learning machines for a two-class imbalance corporate life cycle prediction, Knowledge-Based Systems **39** (2013), 214–223.
[11] N.V. Chawla, K.W. Bowyer and L.O. Hall, SMOTE: synthetic minority over-sampling technique, Journal of Artificial Intelligence Research **16** (2002), 321–357.
[12] E. Parzen, On estimation of a probability density function and mode, The Annals of Mathematical Statistics **33** (1962), 1065–1076.
[13] C. Xue, F. Li, T. He, G.P. Liu, Y. Li and X. Zhang, Classification of real and pseudo microRNA precursors using local structure-sequence features and support vector machine, BMC Bioinformatics **6** (2005), 310.
[14] Q. Zuo, M. Guo, Y. Liu and J. Wang, A classification method for class imbalanced data and its application on bioinformatics, Chinese Journal of Computer Research and Development **17** (2010), 1407–1414
[15] J. Hertel, I.L. Hofacker and P.F. Stadler, SnoReport: computational identification of snoRNAs with unknown targets, Bioinformatics **24** (2008), 158–164.
[16] R. Fletcher, Practical Methods of Optimization, Constrained Optimization, John Wiley & Sons Inc, New Jersey, 1981.
[17] H. Yu, J. Ni, S. Xu, B. Qin and H. Jv, Estimating harmfulness of class imbalance by scatter matrix based class separability measure, Intelligent Data Analysis **18** (2014), 203–216.
[18] C.C. Chang and C.J. Lin, LIBSVM: a library for support vector machines, ACM Transactions on Intelligent Systems and Technology **2** (2011), 27.