Imbalanced classification using support vector machine ensemble

Article *in* Neural Computing and Applications · March 2011

DOI: 10.1007/s00521-010-0349-9 · Source: DBLP

3 authors, including:

Jiang Tian
Dalian University of Technology
6 PUBLICATIONS **106** CITATIONS
SEE PROFILE

Hong Gu
Dalian University of Technology
95 PUBLICATIONS **885** CITATIONS
SEE PROFILE

Neural Comput & Applic (2011) 20:203–209
DOI 10.1007/s00521-010-0349-9

ORIGINAL ARTICLE

Imbalanced classification using support vector machine ensemble

Jiang Tian · Hong Gu · Wenqi Liu

Received: 25 July 2009 / Accepted: 10 February 2010 / Published online: 3 March 2010

Abstract Imbalanced data sets often have detrimental effects on the performance of a conventional support vector machine (SVM). To solve this problem, we adopt both strategies of modifying the data distribution and adjusting the classifier. Both minority and majority classes are resampled to increase the generalization ability. For minority class, an one-class support vector machine model combined with synthetic minority oversampling technique is used to oversample the support vector instances. For majority class, we propose a new method to decompose the majority class into clusters and remove two clusters using a distance measure to lessen the effect of outliers. The remaining clusters are used to build an SVM ensemble with the oversampled minority patterns, the SVM ensemble can achieve better performance by considering potentially suboptimal solutions. Experimental results on benchmark data sets are provided to illustrate the effectiveness of the proposed method.

Keywords Support vector machines · Imbalanced classification · Resample · SMOTE · Ensemble learning

1 Introduction

Class imbalance is prevalent in many real-world domains, such as fraud detection, risk management, medical diagnosis, text classification and information retrieval [1]. For imbalanced data sets, the number of instances in one class outnumbers the number of instances in the other class. Rare objects are typically harder to identify than common objects, and most machine learning algorithms have many difficulties in dealing with rarity, it is important to study the class imbalance problem [2]. Support vector machines (SVMs) are learning machines that can perform classification and regression tasks [3]. Based on sound theoretical principles, SVMs provide better generalization performance than other algorithms. In a conventional SVM classifier, however, a highly imbalanced distribution of data usually brings about poor classification accuracy for unseen samples from the minority class, because the classifier may be strongly biased toward the majority class. SVMs tend to learn how to predict the majority class in particular, although they can get higher predictive accuracies without considering the minority class, this good performance can be identified as meaningless.

In recent years, the machine learning community has addressed the issue of class imbalance mainly in two different ways [1, 2]. The first way involves modifying the classifiers to adapt them to the imbalanced data sets, for example, ensemble learning [4–6], cost-sensitive learning [7, 8] and one class learning [9, 10]. The second classifier-independent way involves balancing the original data set, for example, oversampling [11, 12] and undersampling [13–15]. Chawla et al. [16] proposed the synthetic minority oversampling technique (SMOTE) algorithm in which the minority class was oversampled by taking each minority class sample and introducing new synthetic examples joining any or all the minority class nearest neighbors. Further, algorithms that combine SMOTE and other learning methods have also been applied to solve the class imbalance problem [17–19].

A combination of the two unrelated approaches can perform better than either alone, so we propose a new system integrating both resampling and ensemble techniques. To the

J. Tian (✉) · H. Gu · W. Liu
School of Electronic and Information Engineering,
Dalian University of Technology, Dalian, China
e-mail: tianjiang@gmail.com

best of our knowledge, this is the first time to make use of both the techniques, thus our research aims to provide a more effective method to solve the imbalanced classification problem. In general terms, our approach is different from previous studies in several ways, which enable our technique more robust and effective. First, the support vector instances of minority class are estimated by an one-class support vector machine (OCSVM) model. Only, the support vector instances are oversampled to avoid introducing too many outliers into the training data set. Second, to balance the two classes, we propose a new method to decompose the majority class into clusters. Calculating the distances from each cluster to the minority examples, the nearest and the farthest clusters will be removed if the imbalance rate is enough high. Third, using the remain clusters of majority class and oversampled minority patterns, we build an ensemble of SVMs for improved imbalanced classification. Experimental results show that our approach gives an impressive improvement in prediction performance.

2 Conventional SVM and imbalanced classification

The main idea behind SVM is to find the optimal margin hyperplane that best separates the data paints into two classes based on the structural risk minimization principle [3]. Given l training data points

$$\{(x_i, y_i)\}_{i=1}^{l}, \quad x_i \in R^N, \quad y_i \in \{-1, 1\}.$$

For a soft margin method, SVM requires the solution of the following optimization problem:

$$\begin{aligned} \min \quad & \tfrac{1}{2}||w||^2 + C\sum_i \xi_i, \\ \text{s.t.} \quad & y_i((w \cdot \phi(x_i)) + b) \geqslant 1 - \xi_i, \\ & \xi_i \geqslant 0, \quad i = 1, \ldots, l. \end{aligned} \tag{1}$$

where C is a trade off between a large margin and a small error penalty, ξ_i are slack variables, which measure the degree of misclassification of the datum x_i. A Lagrangian method is used to solve the above problem. The dual of (1) can be shown to be the following optimization problem:

$$\begin{aligned} \min \quad & \tfrac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j K(x_i, x_j) - \sum_i \alpha_i, \\ \text{s.t.} \quad & \sum_i \alpha_i y_i = 0, \\ & 0 \leqslant \alpha_i \leqslant C, \end{aligned} \tag{2}$$

where α is the Lagrangian parameter, $K(x_i, x_j)$ is a kernel function. The decision function is of the form:

$$y = \operatorname{sgn}\left(\sum_i y_i \alpha_i K(x, x_i) + b_0\right). \tag{3}$$

When the class distribution is too skewed, SVMs will run into difficulties [20, 21]. It can be shown from (1) that minimizing the first term on the right hand side $\|w\|^2/2$ is equivalent to maximizing the margin, while minimizing the second term $\sum_i \xi_i$ means minimizing the associated error. The constant parameter C is the trade off between maximizing the margin and minimizing the error. If C is not very large, SVM simply learns to classify everything as negative because that makes the margin the largest, with zero cumulative errors on the abundant negative examples [20]. The corresponding trade off is only the small amount of cumulative error on the positive examples, which do not count for much. Thus, SVM fails in situations with a high degree of imbalance. Besides, SVM tends to produce an insignificant model by almost predicting the majority class, so the classification result is obviously not desired. The hyperplane learned by (3) will be far from the outlier examples, therefore, SVM learns a boundary that is skewed towards the outlier class.

3 Proposed method

The intuition behind our approach is to combine ensemble learning algorithms with resampling techniques. This combination will allow us to lessen the imbalance ratio and, therefore, make the learning task more tractable. The framework of proposed method is showed in Fig. 1. The training patterns include two parts which belong to majority class and minority class, respectively, the detail procedure is as follows:

First, for the minority class, we apply SMOTE algorithm to create new synthetic examples from the support vector instances of an OCSVM model. Without adding too much

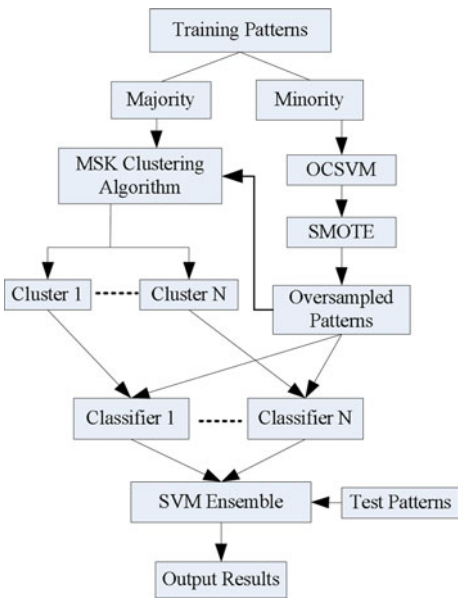

Fig. 1 The detailed procedure of the proposed method

noise into the data set, the minority examples will be oversampled.

Second, we decompose the majority examples into k partitions using a novel method named Middle Score K-means (MSK) clustering algorithm (Sect. 3.2 will give a detailed description). After the process of decomposition, we calculate the distances from the disjunctive clusters to the oversampled minority patterns, then remove two clusters with farthest and nearest distances.

Third, we combine the resampled minority examples with the remaining clusters of majority examples. Thus, each remaining cluster consists of the oversampled minority examples and undersampled majority instances.

The final step is to train SVMs independently on every subset of the training data set and combine all the constituent SVMs using a majority vote method. In this step, several predictors are combined to achieve increased predictive power. All the part classifiers make their predictions, and the SVM ensemble provides the output results of the test data set.

3.1 Using SMOTE on the support vector instances

Since minority examples are usually much fewer than majority examples, it is important that any imbalance reduction procedure should not remove a minority example from the original data. However, the rare examples may include outliers or noises that can be copied by oversampling algorithms. Under this condition, we can assume that oversampling on entire minority examples may introduce outliers into the training data set and degrade the performance of the resulting classifiers. To get off the horns of this dilemma, we pick out only support vector instances of an OCSVM model, which is trained on the entire minority patterns.

One-class support vector machine is an unsupervised learning algorithm that trains on unlabeled samples. The purpose of one class classification is to distinguish one class of target samples from all other class samples [22], for example, the outliers. The optimization is to find the maximized margin and propose an algorithm that estimates a binary function. The maximum margin from the origin is achieved by solving the following quadratic optimization problem:

$$\begin{aligned} \min \quad & \tfrac{1}{2}\|w\|^2 + \tfrac{1}{v\ell}\sum_i \xi_i - \rho, \\ \text{s.t.} \quad & w \cdot \phi(x_i) \geq \rho - \xi_i \\ & \xi_i \geq 0, \quad i = 1, \ldots, l. \end{aligned} \tag{4}$$

The decision function can be shown to have the SV explanation:

$$f(x) = \operatorname{sgn}\left(\sum_i \alpha_i K(x_i, x) - \rho\right), \tag{5}$$

where the coefficients are found as the solution of the dual problem:

$$\begin{aligned} \min \quad & \tfrac{1}{2}\sum_{ij} \alpha_i \alpha_j K(x_i, x_j), \\ \text{s.t.} \quad & 0 \leq \alpha_i \leq 1/(vl), \\ & \sum_i \alpha_i = 1. \end{aligned} \tag{6}$$

In (6), all patterns $\{x_i : \alpha_i > 0\}$ are called support vectors. Note that $v \in (0, 1)$ in (4) is a parameter that controls the tradeoff between maximizing the distance from the origin and containing most of the data in the region created by the hyperplane. It is proved in [22] that v is an upper bound on the fraction of outliers and a lower bound on the fraction of support vectors.

Chawla et al. [16] indicated that duplicating the minority class will not lead to the decision boundary to move to the majority class but make the classifier more specific and possibly over-fit. SMOTE works by creating new synthetic minority samples between two existing minority samples. For each minority sample, SMOTE identifies its nearest positive neighbors and generates new positive synthetic samples randomly along the hyperline connecting the sample and its neighbors. SMOTE makes the distribution of minority class denser and more well defined. In this research, SMOTE is used to resample the support vector instances, the minority class will be oversampled by adding these new synthetic instances. SMOTE will provide more related minority class instances to learn from, thus allowing an SVM classifier to carve broader decision regions, leading to more coverage of the minority class.

3.2 MSK clustering algorithm

Proposed MSK clustering algorithm is based on the K-means method, which is one of the most popular and well-known clustering algorithms [23]. A major advantage of the K-means algorithm is its computational simplicity, which makes it an attractive candidate for various applications. Our purpose is to decompose the majority examples into partitions and omit the unnecessary parts based on the distances to minority examples.

Given a set of observations $\{x_1, x_2, \ldots, x_n\}$, where each observation is a d-dimensional real vector, the K-means clustering algorithm aims at partitioning this set into k partitions ($k < n$) to minimize total intra-cluster variance or the squared error function:

$$V = \sum_{i=1}^{k} \sum_{x_j \in S_i} \| x_j - \mu_i \|^2, \tag{7}$$

where S_i, $i = 1, 2, \ldots, k$, are k clusters, and $\|x_j - \mu_i\|^2$ is a chosen distance measure between a data point x_j and the cluster center μ_i. In our approach, the Mahalanobis distance is chosen as the distance measure, which is a useful

way of deciding the similarity of an unknown sample set to a known one. Differing from Euclidean distance, the Mahalanobis distance considers the correlations of the data set and is scale invariant, in other words, it is not dependent on the scale of measurements.

The MSK clustering algorithm includes two steps. In the first step, the K-means algorithm starts by partitioning the training points into k initial sets, either at random or by using heuristic data. The parameter k is determined by the truncation of the current imbalance rate, and the mean point of each set is also calculated. The algorithm builds a new partition by associating each point with the closest centroid. The centroids are recalculated for calculating the new clusters. The algorithm will be looped by alternate application of these two steps until the points no longer switch clusters, or the centroids are no longer changed.

In the second step, the distances from the k clusters to the resampled minority examples are calculated. We calculate the Mahalanobis distances between the cluster centroids and the mean of the oversampled minority examples, these distances reflect the degree of similarity. The K-means algorithm is sensitive to outliers and noise, the clusters containing outliers will behave abnormally compared with the other clusters. After calculating the distances, we treat the distances as scores for estimating the abnormality. The clusters with largest and smallest scores are assumed to have large impacts on the decision function. Our strategy is to keep the clusters with mild scores, which means that two clusters with largest and smallest distances will be artificially removed. Notice that if k is not large, all the clusters will be kept. Thus, by combining each remaining cluster with the oversampled minority examples, we get the training data sets for further classification.

4 Experiments

4.1 Experimental description

We chose nine different imbalanced data sets for the experiment, obtained from the UCI Repository for Machine Learning. All the data sets were converted to binary class problems to make their imbalance ratio as high as possible. Table 1 lists the data sets, with some relevant statistics, such as the number of positive and negative examples, their ratios, and the name of the positive class. Because the information about the UCI data sets is public, we do not report it here.

For imbalanced data, the overall classification accuracy is not a good metric, because a trivial classifier that predicts every case as the majority class could achieve high accuracy in skewed domains. We use the F-measure to measure the overall performance, defined as the compromise between recall and precision, of the algorithms studied. The exact definitions of recall (R) and precision (P) were first introduced in the information retrieval community. The F-measure is defined as

$$F = 2 \times \left(\frac{R \times P}{R + P} \right), \tag{8}$$

it is a harmonic mean between recall and precision.

The experimental software was developed based on Libsvm [24]. For all the SVM classifiers, we used an RBF kernel. Weka [25] was chosen to carry out the different algorithms. The training and test data sets were selected randomly from the original data sets. The experiment had been carried out for 10 times, produced predictions with different performance values each time. For all the methods, the parameters were optimized using a 10-fold cross validation strategy. Our method combines both resampling and ensemble techniques, so it is necessary to compare against both kinds of approaches alone. The details will be described in the following section.

Table 1 Description of the experimental data sets

Data set	Pos class	Ratio	Training		Test	
			Pos	Neg	Pos	Neg
B-cancer	Recur-events	2.36	28	67	57	134
Car	vgood	25.2	22	554	44	1,109
Glass	Headlamps	6.38	9	61	20	124
Balance-scale	B	11.76	16	192	33	384
Letter	Letter-T	24.13	265	6,401	531	12,803
Hypothyroid	Hypothyroid	11.96	97	1,160	194	2,321
Soybean	Brown-spot	6.42	30	197	62	394
Segment	Sky	6	110	660	220	1,320
Sick	Sick	15.33	77	1,180	154	2,361

4.2 Comparison with resampling methods

We first compare the performance of our method with SVM-based methods: standard SVM, SVM with undersampling, and SVM with SMOTE. For undersampling we produce a random subsample of the data set. For SMOTE, the minority class is oversampled to have the same quantity with the majority class. The results are summarized in Tables 2, 3 and 4. Each performance measure is gained by running the experiment 10 times with different training and test data sets. The measurements are averaged and expressed with the standard error of the mean.

From these tables, we find that the recalls of our method and SVM with undersampled data are the best for recall performance measure. On Car data set, our method and SVM with undersampling have the same recall of 0.977, and each method perform best on the other different four

Table 2 Recall comparisons among SVM, SVM with undersampling, SVM with oversampling and our method

R	SVM	Undersampling	Oversampling	Our method
B-cancer	0.316 ± 0.021	0.404 ± 0.011	0.386 ± 0.016	0.509 ± 0.011
Car	0.932 ± 0.006	0.977 ± 0.003	0.907 ± 0.008	0.977 ± 0.005
Glass	0.7 ± 0.035	0.75 ± 0.022	0.7 ± 0.017	0.65 ± 0.017
Balance-scale	0.091 ± 0.024	0.788 ± 0.011	0.091 ± 0.011	0.879 ± 0.022
Letter	0.947 ± 0.003	0.985 ± 0.006	0.957 ± 0.002	0.972 ± 0.001
Hypothyroid	0.046 ± 0.013	0.902 ± 0.008	0.392 ± 0.001	0.495 ± 0.011
Soybean	0.919 ± 0.021	0.952 ± 0.027	0.903 ± 0.026	0.887 ± 0.018
Segment	0.077 ± 0.006	0.218 ± 0.003	0.241 ± 0.004	0.523 ± 0.006
Sick	0.065 ± 0.004	0.545 ± 0.011	0.247 ± 0.009	0.688 ± 0.004

Table 3 Precision comparisons among SVM, SVM with undersampling, SVM with oversampling and our method

P	SVM	Undersampling	Oversampling	Our method
B-cancer	0.353 ± 0.013	0.307 ± 0.017	0.361 ± 0.005	0.475 ± 0.009
Car	0.103 ± 0.008	0.084 ± 0.001	0.102 ± 0.003	0.124 ± 0.004
Glass	0.933 ± 0.022	0.536 ± 0.014	0.875 ± 0.006	0.929 ± 0.005
Balance-scale	0.2 ± 0.032	0.1 ± 0.009	0.6 ± 0.008	0.14 ± 0.015
Letter	0.979 ± 0.002	0.603 ± 0.005	0.973 ± 0.005	0.977 ± 0.002
Hypothyroid	0.6 ± 0.009	0.13 ± 0.013	0.589 ± 0.002	0.676 ± 0.012
Soybean	0.64 ± 0.013	0.44 ± 0.019	0.727 ± 0.009	0.688 ± 0.019
Segment	0.944 ± 0.001	0.96 ± 0.005	0.981 ± 0.004	0.991 ± 0.003
Sick	0.303 ± 0.001	0.137 ± 0.006	0.184 ± 0.007	0.185 ± 0.013

data sets, respectively. As expected, the undersampling technique improves the recall rate over the original data or SMOTE. However, SVM with undersampling obtains high recall at the expense of the precision rate, while SVM with SMOTE gives much better precision. For precision performance measure, our method performs best on five data sets, while SVM with SMOTE obtains the highest mean value, because the precision of SVM with SMOTE on Balance-scale data set is much higher than those of the other methods. For *F*-measure measure, our method performs best and SVM with SMOTE performs second best, while the other two methods show almost the same performance.

Analyzing the above results, we see that the mean recall of our method is superior to that of all the other methods. The mean recall of our method is 0.731. It is better than SVM with undersampling, which has a mean recall of 0.725, the second-best performance. SVM with oversampling has about higher mean precision than our method, the values are 0.599 and 0.576, respectively. The *F*-measure is the combination of recall and precision, our method reaches the best *F*-measure (Table 4). We believe that our method has both the advantages of undersampling and oversampling techniques, while overcomes their individual disadvantages to some extent.

Table 4 *F*-measure comparisons among SVM, SVM with undersampling, SVM with oversampling and our method

F-measure	SVM	Undersampling	Oversampling	Our method
B-cancer	0.329 ± 0.014	0.343 ± 0.008	0.371 ± 0.011	0.491 ± 0.006
Car	0.183 ± 0.005	0.154 ± 0.005	0.182 ± 0.007	0.213 ± 0.005
Glass	0.798 ± 0.023	0.625 ± 0.021	0.776 ± 0.013	0.764 ± 0.008
Balance-scale	0.125 ± 0.025	0.177 ± 0.019	0.155 ± 0.013	0.241 ± 0.011
Letter	0.962 ± 0.002	0.745 ± 0.006	0.961 ± 0.007	0.973 ± 0.001
Hypothyroid	0.081 ± 0.015	0.223 ± 0.017	0.466 ± 0.001	0.567 ± 0.015
Soybean	0.754 ± 0.008	0.601 ± 0.016	0.803 ± 0.017	0.773 ± 0.021
Segment	0.137 ± 0.001	0.352 ± 0.007	0.386 ± 0.009	0.685 ± 0.005
Sick	0.104 ± 0.002	0.216 ± 0.008	0.207 ± 0.006	0.291 ± 0.008

In addition, it is interesting that the performances on the Car, Glass and Letter data sets remain nearly unaffected. These facts indicate that resampling techniques are data dependent. Further, for the B-cancer data set, the performances of both methods are worse than those of the Letter data set, even though it has the lowest imbalance ratio of 2.36. This suggests the imbalance ratio itself do not always make the learning problem difficult. Moreover, we see that some data sets are almost not affected by the imbalance in the data sets, especially B-cancer and Car. This finding suggests that these data sets are already ideal for the learning, or the resampling techniques cannot improve the decision hyperplane in the feature space. Variations of the original data distribution could also yield to suboptimal performance. SVM with resampling techniques shows different performance for various data sets. For the Balance-scale and Sick data sets, SVM with undersampling works much better compared with the other data sets. For the B-cancer, Soybean and Segment data sets, SVM with SMOTE performs better under the precision measure.

Observing the details, our method outperforms all the other methods on most of the data sets. As a matter of fact, there are only two data sets (Glass and Soybean) for which SVM with oversampling is the ideal classifier. The results also indicate that SVM with oversampling is the second-best classifier, and the other two methods have almost the same lesser performance over all the data sets. The most remarkable change of *F*-measure appears on the Hypothyroid and Segment data sets, where the performance of the standard SVM has been remarkably improved by all the other methods. The difference is unremarkable in Glass and Letter, which means the imbalance has slight impact on these data sets. These results also show that our method and the other resampling techniques are data dependent.

4.3 Comparison with ensemble methods

Ensemble techniques with bagging or boosting have been shown to improve the classification performance of real SVMs [26], Bagging [27], AdaboostM1 [28] and MultiBoosting [29] are three classical ensemble techniques. Under the same experimental conditions, the mean performance of different ensemble techniques is shown in Figs. 2, 3 and 4.

From these figures, it is interesting to see that all the methods have similar performance on Car, Glass, Letter and Soybean data sets. Among these four data sets, the proposed method performs best on two data sets and AdaboostM1 performs best on the other two data sets. The proposed method performs best on other four data sets, they are B-cancer, Balance-scale, Hypothyroid and Segment. So, the

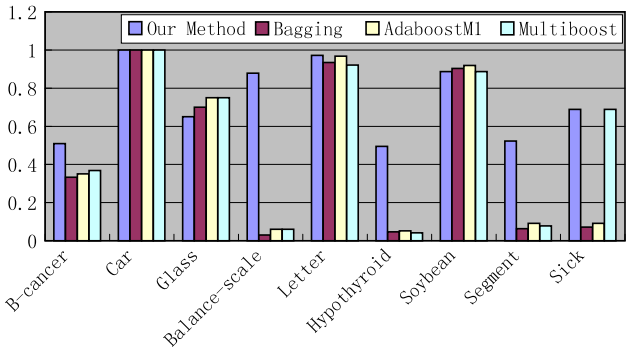

Fig. 2 Comparison of recall of Bagging, AdaboostM1, MultiBoosting, proposed method for nine data sets

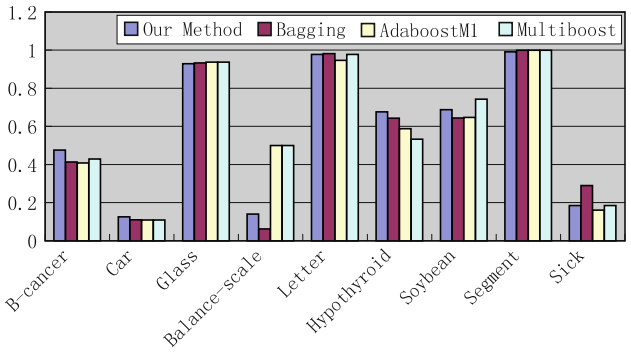

Fig. 3 Comparison of precision of Bagging, AdaboostM1, MultiBoosting, proposed method for nine data sets

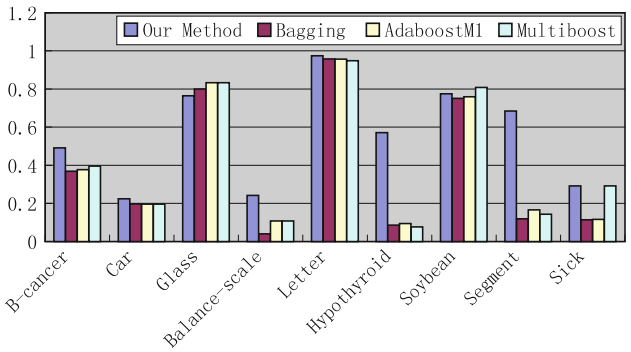

Fig. 4 Comparison of *F*-measure of Bagging, AdaboostM1, MultiBoosting, proposed method for nine data sets

proposed method performs best on six data sets out of nine, suggesting that our method manage to find meaningful ensembles that can be used to identify many majority instances. The mean *F*-measure of our method is higher than those of other methods. In addition, we find that MultiBoosting gives the second-best results, because it obtains the best *F*-measure for Glass and Soybean data sets. We believe that the other methods can improve the performance of SVM classifier, while our method is superior for imbalance learning.

5 Conclusion

In this research, we proposed a novel resampling approach and build an SVM ensemble for tackling problems associated with imbalanced data sets. Through theoretical justifications and experimental studies, the proposed method is demonstrated to be effective. In particular, our method mainly improves the performance in two ways. First, the minority examples are trained using an OCSVM model, the SMOTE technique is applied on the support vector examples in order not to add too many outliers into the data set. Second, after removing the clusters farthest or nearest to the minority class, the novel MSK clustering algorithm decomposes the majority class into clusters for improving the generalization ability of the SVM ensemble. The experimental results show the effectiveness of our method, also suggest that all the methods are data dependent. In future work, we will apply our approach to many other data sets.

References

1. Chawla NV, Japkowicz N, Kotcz A (2004) Editorial: special issue on learning from imbalanced data sets. SIGKDD Explor 6(1):1–6
2. Weiss GM (2004) Mining with rarity: a unifying framework. SIGKDD Explor 6(1):7–19
3. Vapnik VN (2000) The nature of statistical learning theory. Springer, New York
4. Zhao XM, Li X, Chen L, Aihara K (2008) Protein classification with imbalanced data. Proteins 70(4):1125–1132
5. Wu MR, Ye JP (2009) A small sphere and large margin approach for novelty detection using training data with outliers. IEEE Trans Pattern Anal Mach Intell 31(11):2088–2092
6. Li X, Wang L, Sung E (2008) Adaboost with svm-based component classifiers. Eng Appl Artif Intell 21(5):785–795
7. Zhou ZH, Liu XY (2006) Training cost-sensitive neural networks with methods addressing the class imbalance problem. IEEE Trans Knowl Data Eng 18(1):63–77
8. Sun Y, Kamel MS, Wong AKC, Wang Y (2007) Cost-sensitive boosting for classification of imbalanced data. Pattern Recognit 40(12):3358–3378
9. Gardner AB, Krieger AM, Vachtsevanos G, Litt B (2006) One-class novelty detection for seizure analysis from intracranial eeg. J Mach Learn Res 7(7):1025–1044
10. Giacinto G, Perdisci R, Del Rio M, Roli F (2008) Intrusion detection in computer networks by a modular ensemble of one-class classifiers. Inf Fusion 9(1):69–82
11. Liao TW (2008) Classification of weld flaws with imbalanced class data. Expert Syst Appl 35(3):1041–1052
12. Guo H, Viktor H (2004) Learning from imbalanced data sets with boosting and data generation: the databoost-im approach. SIGKDD Explor 6(1):30–39
13. Yen SJ, Lee YS (2009) Cluster-based under-sampling approaches for imbalanced data distributions. Expert Syst Appl 36(3):5718–5727
14. Kubat M, Matwin S (1997) Addressing the curse of imbalanced training sets: one-sided selection. In: Proceedings of the 14th international conference on machine learning, Nashville, pp 179–186
15. Wilson DR, Martinez TR (2000) Reduction techniques for instance-based learning algorithms. Mach Learn 38(3):257–286
16. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) Smote: Synthetic minority over-sampling technique. J Artif Intell Res 16(6):321–357
17. Liu Y, An A, Huang X (2006) Boosting prediction accuracy on imbalanced datasets with svm ensembles. In: 10th Pacific-Asia conference on advances in knowledge discovery and data mining. Springer, Singapore, pp 107–118
18. Wang HY (2008) Combination approach of smote and biased-svm for imbalanced datasets. In: Proceedings of the international joint conference on neural networks. Institute of Electrical and Electronics Engineers Inc., Hong Kong, pp 228–231
19. Yoon K, Kwek S (2007) A data reduction approach for resolving the imbalanced data issue in functional genomics. Neural Comput Appl 16(3):295–306
20. Akbani R, Kwek S, Japkowicz N (2004) Applying support vector machines to imbalanced datasets. In: Proceedings of the 15th European conference on machine learning. Springer, Pisa, pp 39–50
21. Wu TF, Lin CJ, Weng RC (2004) Probability estimates for multi-class classification by pairwise coupling. J Mach Learn Res 5(5):975–1005
22. Scholkopf B, Platt JC, Shawe-Taylor J, Smola AJ, Williamson RC (2001) Estimating the support of a high-dimensional distribution. Neural Comput 13(7):1443–1471
23. Lloyd SP (1982) Least-squares quantization in pcm. IEEE Trans Inf Theory 28(2):129–137
24. Chang CC, Lin CJ (2001) Libsvm: a library for support vector machines. Software available at http://www.csie.ntu.edu.tw/cjlin/libsvm
25. Witten IH, Frank E (2005) Data mining: practical machine learning tools and techniques. Morgan Kaufmann Publisers, San Fransisco
26. Kim HC, Pang S, Je HM, Kim D, Bang SY (2003) Constructing support vector machine ensemble. Pattern Recognit 36(12):2757–2767
27. Breiman L (1996) Bagging predictors. Mach Learn 24(2):123–140
28. Freund Y, Schapire RE (1996) Experiments with a new boosting algorithm. In: Proceedings of the thirteenth international conference on machine learning. Morgan Kauffman, San Francisco, pp 148–156
29. Webb GI (2000) Multiboosting: a technique for combining boosting and wagging. Mach Learn 40(2):159–196