Hindawi Publishing Corporation
Abstract and Applied Analysis
Volume 2013, Article ID 196256, 6 pages
http://dx.doi.org/10.1155/2013/196256

Research Article

A Cost-Sensitive Ensemble Method for Class-Imbalanced Datasets

Yong Zhang and Dapeng Wang

School of Computer and Information Technology, Liaoning Normal University, No. 1, Liushu South Street, Ganjingzi, Dalian, Liaoning 116081, China

Correspondence should be addressed to Yong Zhang; zhyong@lnnu.edu.cn

Received 28 December 2012; Accepted 25 March 2013

Academic Editor: Jianhong (Cecilia) Xia

In imbalanced learning methods, resampling methods modify an imbalanced dataset to form a balanced dataset. Balanced data sets perform better than imbalanced datasets for many base classifiers. This paper proposes a cost-sensitive ensemble method based on cost-sensitive support vector machine (SVM), and query-by-committee (QBC) to solve imbalanced data classification. The proposed method first divides the majority-class dataset into several subdatasets according to the proportion of imbalanced samples and trains subclassifiers using AdaBoost method. Then, the proposed method generates candidate training samples by QBC active learning method and uses cost-sensitive SVM to learn the training samples. By using 5 class-imbalanced datasets, experimental results show that the proposed method has higher area under ROC curve (AUC), F-measure, and G-mean than many existing class-imbalanced learning methods.

1. Introduction

In the classification problem field, the scenario of imbalanced data sets appears when the number of samples that represent the different classes is very different among them [1]. Class-imbalanced problems widely exist in the fields of medical diagnosis, fraud detection, network intrusion detection, science and engineering problems, and so on. We consider the binary-class-imbalanced data sets, where there is only one positive (minority) class and one negative (majority) class. Most of data are in the majority class, and little data are in the minority class. Many traditional classification methods tend to be overwhelmed by the majority class and ignore the minority class. The classification performance for the positive class becomes unsatisfactory.

It is important to select the suitable training data for classification in the class-imbalanced classification problem. Resampling is one of the effective techniques for adjusting the size of training sets. Many resampling methods are used to reduce or eliminate the extent of data set imbalance, such as oversampling the minority class, undersampling the majority class, and the combination of both methods. Resampling techniques can be used with many base classifiers, such as support vector machine (SVM), C4.5, Naïve Bayes classifier, and AdaBoost, to address the class-imbalanced problem. So, it provides a convenient and effective way to deal with imbalanced learning problems using standard classifiers [2]. Additionally, modified learning algorithmic solutions are the effective approaches to the imbalanced data classification problem. These solutions are obtained by modifying existing learning algorithms so that they can deal with imbalanced problems effectively. Integrated approach, cost-sensitive learning, feature selection, and single-class learning belong to the solutions. Cost-sensitive learning deals with class imbalance by incurring different costs for the two classes and is considered an important type of methods to handle class imbalance. The difficulty with cost-sensitive classification is that costs of misclassification are often unknown [3].

Although the existing imbalance-learning methods applied for normal SVMs can solve the problem of class imbalance, they can ignore potential useful information in major samples, and probably lead to overfitting problem.

This paper presents a cost-sensitive ensemble method. The proposed method uses AdaBoost method to train subclassifiers according to the ratio of imbalanced samples, integrates these sub-classifiers into a classifier, and uses cost-sensitive SVM to train the candidate data selected by a query-by-committee (QBC) algorithm.

The rest of the paper is organized as follows. Following the introduction, Section 2 presents a comprehensive study on the class-imbalanced problem and discusses the existing class-imbalanced solutions. Section 3 simply introduces cost-sensitive SVM. Section 4 proposes a cost-sensitive ensemble method for class-imbalanced data sets. In Section 5, we apply a statistical test to compare the performance of the proposed method with the existing methods. Finally, Section 6 concludes this paper.

2. Related Work

Many techniques are proposed to solve classification problems based on imbalanced data sets. There are two major categories of techniques developed to address the class-imbalance issue. One is resampling and the other is modified learning algorithmic solutions [4].

Resampling is one of the effective techniques for adjusting the size of a training dataset. In general, it can be further divided into undersampling approach and over-sampling approach. Undersampling uses only some samples of the majority class to reduce the data size and removes samples of the majority class to balance a data set. So the risk is that the reduced sample set may not represent the full characteristics of the majority class. There are many studies which discuss under-sampling methods. For example, Kim [5] proposes an under-sampling method based on a self-organizing map (SOM) neural network to obtain sampling data which retains the original data characteristics. Yen and Lee [6] present a cluster-based under-sampling approach for selecting the representative data as training data. The proposed method improves the classification accuracy for the minority class. Aiming at the deficiency of under-sampling where many majority-class samples are ignored, Liu et al. [7] propose two effective informed under-sampling methods, EasyEnsemble and BalanceCascade. EasyEnsemble method samples several subsets from the majority-class, trains a learner using each of them, and combines the outputs of those learners. BalanceCascade method trains the learners sequentially. In each step of BalanceCascade, the majority class samples which are correctly classified by the current trained learners are removed from further consideration.

The over-sampling approach is to add more new data instances to the minority class to balance a data set. These new data instances can either be generated by replicating the data instances of the minority class or by applying synthetic methods. However, over-sampling often involves making exact copies of samples which may lead to overfitting [8]. synthetic minority oversampling technique (SMOTE) [1] is an intelligent over-sampling method using synthetic samples. SMOTE method adds new synthetic samples to the minority class by randomly interpolating pairs of the closest neighbors in the minority class. SMOTEBoost algorithm [9] combines SMOTE technique and the standard boosting procedure. It utilizes SMOTE for improving the accuracy over the minority class and utilizes boosting not to sacrifice accuracy over the entire data set. Wang et al. [10] propose an adaptive over-sampling technique based on data density (ASMOBD), which can adaptively synthesize different number of new samples around each minority sample according to its level of learning difficulty. Gao et al. [11] propose probability density function estimation based on over-sampling approach for two class-imbalanced classification problems.

At the algorithmic level, the solutions mainly include cost-sensitive learning, integrated approach, and modified algorithms. Many cost-sensitive learning methods have been proposed [12, 13]. A common strategy of these methods is to intentionally increase the weights of samples with higher misclassification cost in the boosting process. However, misclassification costs are often unknown, and a cost-sensitive classifier may result in over-fitting training. Sun et al. [14] investigate cost-sensitive boosting algorithms for advancing the classification of imbalanced data and propose three cost-sensitive boosting algorithms by introducing cost items into the learning framework of AdaBoost. Guo and Viktor [15] propose a modified boosting procedure, DataBoost, to solve the imbalanced problem. DataBoost combines the boosting and ensemble-based learning algorithms. In terms of modified algorithms, several specific attempts using SVMs have been made at improving their class prediction accuracy in the case of class imbalances [16, 17]. The results obtained with such methods show that SVMs have the particular advantage of being able to solve the problem of skewed vector spaces, without introducing noise. Wang and Japkowicz [13] combine modifying the data distribution approach and modifying the classifier approach in class-imbalanced problem and use support vector machines with soft margins as the base classifier to solve the skewed vector spaces problem.

In addition, Wang et al. [18] develop two models to yield the feature extractors and propose a method for extracting minimum positive and maximum negative features for imbalanced binary classification. Based on the divide-and-conquer principle, the scalable instance selection approach OligoIS is proposed in [19] for class-imbalanced data sets. OligoIS can deal with the class-imbalanced problem that is scalable to data sets with many millions of instances and hundreds of features.

3. Cost-Sensitive SVM

SVM has been widely used in many application areas of machine learning. The goal of the SVM-learning algorithm is to find a separating hyperplane that separates these data points into two classes. In order to find a better separation of classes, the data are first transformed into a higher-dimensional feature space. However, regular SVM is invalid to the imbalanced data sets. For imbalanced data sets, the learned boundary is too close to the minority samples, so SVM should be biased in a way that will push the boundary away from the positive samples [16]. Using different error costs for the positive and negative classes, SVM can be extended to the cost-sensitive setting by introducing an additional parameter that penalizes the errors asymmetrically.

Consider that we have a binary classification problem, which is represented by a data set $\{(x_1, y_1), (x_2, y_2), \ldots, (x_l, y_l)\}$, where $x_i \subset \mathfrak{R}^k$ represents a k-dimensional data point and $y_i \in \{+1, -1\}$ represents the class of that data point, for $i = 1, \ldots, l$. Let $I_+ = \{i : y_i = +1\}$ and $I_- = \{i : y_i = -1\}$. The support vector technique requires the solution of the quadratic programming problem as follows [20]:

$$\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C^+ \sum_{i\in I_+} \xi_i + C^- \sum_{i\in I_-} \xi_i \qquad (1)$$

subject to

$$\begin{gathered} y_i (w \cdot \phi(x_i) + b) \geq 1 - \xi_i, \\ \xi_i \geq 0, \quad i = 1, \ldots, l, \end{gathered} \qquad (2)$$

where the training vectors x_i are mapped into a higher-dimensional space by the function ϕ. Parameter C^+ represents the cost of misclassifying the positive sample, and C^- represents the cost of misclassifying the negative sample. The optimal result can be obtained when C^-/C^+ equals the minority-to-majority class ratio. The slack variables $\xi_i > 0$ hold for misclassified samples, and therefore, $\sum_{i=1}^{l} \xi_i$ can be thought of as a measure of the amount of misclassifications. This quadratic-optimization problem can be solved by constructing a Lagrangian representation and transforming it into the following dual problem:

$$\max_{\alpha} W(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \qquad (3)$$

subject to

$$\begin{gathered} 0 \leq \alpha_i \leq C^+ \quad \text{for } i \in I_+, \\ 0 \leq \alpha_i \leq C^- \quad \text{for } i \in I_-, \\ \sum_{i=1}^{l} \alpha_i y_i = 0, \end{gathered} \qquad (4)$$

where α_i is the Lagrangian parameter. Note that the kernel trick $K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$ is used in (3).

4. An Ensemble Method Based on Cost-Sensitive SVM and QBC

This paper presents an ensemble method based on cost-sensitive SVM and QBC, called CQEnsemble, specifically designed for imbalanced data classification. The proposed method applies division and boost techniques to a simple QBC strategy [21, 22] and improves classification precision on the basis of maximizing data balance. In order to overcome the shortages of over-sampling and under-sampling, the CQEnsemble method trains sub-classifiers using AdaBoost algorithm [23] according to the ratio of imbalanced samples and integrates these sub-classifiers into a classifier. AdaBoost can be used in conjunction with many otherlearning algorithms to improve their performance. In this way, the proposed method not only fully uses the minority class information but also feedbacks the different aspects of information of the majority class.

Suppose that an imbalanced dataset contains n samples from the majority class and m samples from the minority class where $n \gg m$. First, the CQEnsemble method divides training data set into m equivalent subsets, where m is greater than or equal to 3. Then, we randomly select two subsets and generate two sub-classifiers as QBCs committees to vote for the other $m - 2$ equivalent subsets. We add samples, in which the vote results are different in two QBC's committees, to candidate data set. It is difficult to decide the category of these samples. So, these samples probably include abundant information. Last, we integrate candidate data set and two selected subsets into new training datasets, train, and get a classifier using cost-sensitive SVM method. Experiments of this paper show that the CQEnsemble method can get comprehensive classification information when the value of m is 5.

Based on the description above, the proposed CQEnsemble method is described as follows.

Algorithm 1 (the CQEnsemble method).

Input. Imbalanced data set D.

Output. An ensemble classifier H.

Step 1. Suppose that the training set is A and the total number of samples is n. Divide A into m $(m \geq 3)$ equivalent subsets randomly, labeled as N_i $(i = 1, 2, \ldots, m)$.

Step 2. Select two subsets randomly and label them as N_i $(i = 1, 2)$ conveniently. For each subset N_i do

Step 2.1. Compute the ratio of the number of majority-class samples to the number of minority-class samples r_i $(i = 1, 2)$.

Step 2.2. Divide the majority-class samples into r_i subsets.

Step 2.3. Merge the minority-class samples and each subset to the training set, and get r_i training sets.

Step 2.4. Classify each training set in Step 2.3 using AdaBoost algorithm, and get r_i weak classifiers H_{ij}, where $j = 1, 2, \ldots, r_i$.

Step 2.5. Regard these weak classifiers H_{ij} as features, and integrate into classifier H_i.

End for

Step 3. Use classifiers H_i $(i = 1, 2)$ to respectively train samples in the rest $m - 2$ subsets, and add samples in which the results are different in two classifiers H_i $(i = 1, 2)$ to new candidate set D_c.

Step 4. Merge two selected subsets N_i $(i = 1, 2)$ to the candidate set D_c, and get a new training set F.

Step 5. Classify data set F using cost-sensitive SVM method, and get a classifier H.

5. Experiment and Analysis

In this section, we first give several evaluation measures for class-imbalanced problem, and then present and discuss, in detail, the results obtained by the experiments carried out in this research.

5.1. Evaluation Measures. Accuracy is an important evaluation metric for assessing the classification performance and guiding the classifier modeling. However, accuracy is not a useful measure for imbalanced data, particularly when the number of instances of the minority class is very small compared with the majority class [24]. For example, if we have a ratio of 1 : 100, a classifier that assigns all instances to the majority class will have 99% accuracy. But this measurement is meaningless to some applications where the learning concern is the identification of the rare cases.

Several measures have been developed to deal with the classification problem with the class imbalance, including *F-measure*, *G-mean*, and *AUC* [25]. Given the number of true positives (TPs), false positives (FPs), true negatives (TNs), and false negatives (FNs), we can obtain the confusion matrix presented in Table 1 after a classification process. We can also define several common measures. The TP rate TPR, recall R, or sensitivity S_n is defined as

$$\text{TPR} = R = S_n = \frac{\text{TP}}{\text{TP} + \text{FN}}. \quad (5)$$

The TN rate TNR or specificity S_p is defined as

$$\text{TNR} = S_p = \frac{\text{TN}}{\text{TN} + \text{FP}}. \quad (6)$$

Precision P is defined as the fraction of relevant instances that are retrieved as follows:

$$P = \frac{\text{TP}}{\text{TP} + \text{FP}}. \quad (7)$$

Based on these measures, other measures have been presented, such as *F-measure* and *G-mean*. *F-measure* is often used in the fields of information retrieval and machine learning for measuring search, document classification, and query classification performance. *F-measure* considers both the precision P and the recall R to compute the score [26]. It can be interpreted as a weighted average of the precision and recall as follows:

$$F\text{-}measure = \frac{2 \times P \times R}{P + R}. \quad (8)$$

G-mean is defined by two parameters called sensitivity S_n and specificity S_p. Sensitivity shows the performance of the positive class, and specificity shows the performance of the negative class. *G-mean* measures the balanced performance of a learning algorithm between these two classes. *G-mean* is defined as

$$G\text{-}mean = \sqrt{S_n \times S_p}. \quad (9)$$

Table 1: Confusion matrix.

	Predicted positive class	Predict negative class
Actual positive class	TP (true positive)	FN (false negative)
Actual negative class	FP (false positive)	TN (true negative)

A receiver operating characteristic (ROC) curve is a graphical plot which depicts the performance of a binary classifier as its discrimination threshold is varied. In an ROC curve, the true positive rate (sensitivity) is plotted in function of the false positive rate (specificity) for different cut-off points. Each point on the ROC curve represents a (sensitivity, specificity) pair corresponding to a particular decision threshold. The ideal point on the ROC curve would be (0, 1); that is, all positive samples are classified correctly, and no negative samples are misclassified as positive. An ROC curve depicts relative trade-offs between benefits (true positives) and costs (false positives) across a range of thresholds of a classification model. However, it is difficult to decide which one is the best method when comparing several classification models. *AUC* is the area under an ROC curve. It has been proved to be a reliable performance measure for imbalanced and cost-sensitive problems [25]. *AUC* provides a single measure of a classifier's performance for evaluating which model is better on average.

5.2. Experimental Results and Analysis. In our experiments, we used 5 data sets to test the performance of the proposed method. These data sets are from the UCI Machine Learning Repository [27]. Information about these data sets is summarized in Table 2. These data sets vary extensively in their sizes and class proportions. We take the minority class as the target class and all the other categories as majority class. When more than two classes exist in the data set, the target class is considered to be positive and all the other classes are considered to be negative. We compared the performance of 5 methods, including AdaBoost, SMOTE [1], SMOTEBoost [9], EasyEnsemble [7], and our proposed CQEnsemble method.

In our experiments, *F-measure*, *G-mean,* and *AUC* are used as metrics. For each data set, we perform a 5-fold cross validation. In each fold four out of five samples are selected to be training set, and the left one out of five samples is testing set. This process repeats 5 times so that all samples are selected in both training set and testing set.

Figure 1 shows the average *F-measure* values of the compared methods. The results show that CQEnsemble has higher *F-measure* than other compared methods on *haberman*, *pima,* and *letter* data sets. EasyEnsemble achieves the highest *F-measure* on *transfusion* data set among these methods, and AdaBoost achieves the highest *F-measure* on *phoneme* data set. The results indicate that CQEnsemble can further improve the *F-measure* metric of imbalanced learning.

TABLE 2: Summary of data sets.

Data set	Total samples	no of attributes	no of positive	no of negative	Ratio (majority/minority)
Haberman	306	3	81	225	2.8
Transfusion	926	4	178	748	4.2
Pima	768	8	268	500	1.9
Phoneme	5404	5	1586	3818	2.4
Letter	20000	16	789	19211	24.3

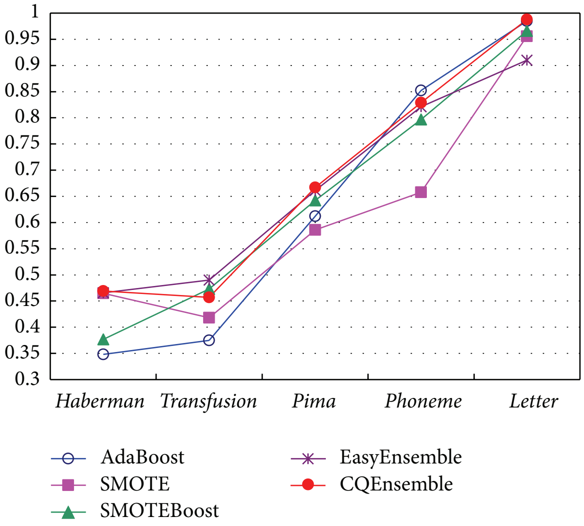

FIGURE 1: *F-measure* of the compared methods.

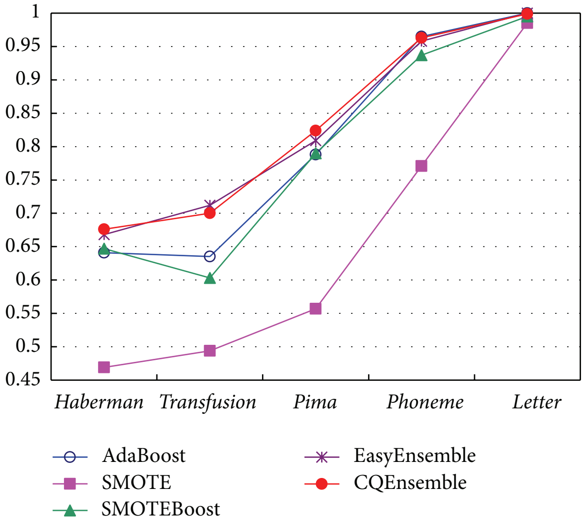

FIGURE 3: *AUC* of the compared methods.

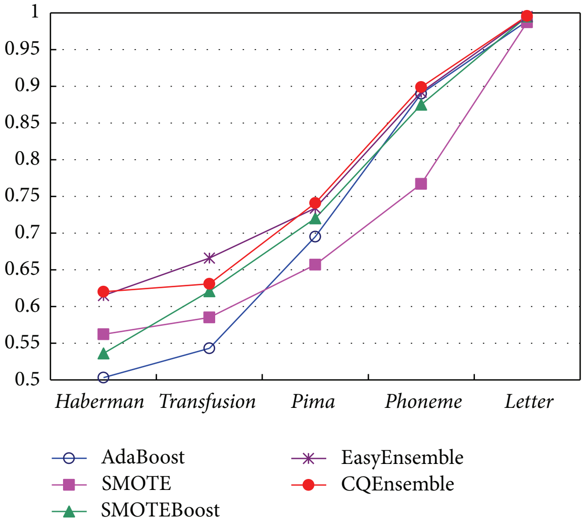

FIGURE 2: *G-mean* of the compared methods.

The average *G-mean* values of the compared methods are summarized in Figure 2. The results show that CQEnsemble has higher *G-mean* than other compared methods on most of datasets, while EasyEnsemble is slightly higher *G-mean* than CQEnsemble on *transfusion* dataset. From Figures 1 and 2, EasyEnsemble has the highest *F-measure* and *G-mean* on *transfusion* dataset among these methods.

Figure 3 shows the *AUC* metric of each method for *haberman*, *transfusion*, *pima*, *phonem,e* and *letter* data sets. The results show that the proposed CQEnsemble method obtains the highest average *AUC* among these compared methods. These methods are equivalent for *letter* data set. After all, SMOTE method is the weakest in 5 methods; EasyEnsemble method is slightly better than AdaBoost, SMOTE, and SMOTEBoost, while CQEnsemble method is better than EasyEnsemble method. The results show that the CQEnsemble method effectively avoids the shortages of resampling methods.

CQEnsemble attains higher average *F-measure*, *G-mean*, and *AUC* than almost all the other methods, except that CQEnsemble is slightly worse comparable to EasyEnsemble with *F-measure*, *G-mean*, and *AUC* on *transfusion* data set. The experimental results imply that the proposed CQEnsemble method is better than AdaBoost, SMOTE, SMOTEBoost, and EasyEnsemble methods on most of data sets. These experiments also indicate that the combination of division-boost method and cost-sensitive learning can further improve the performance of imbalanced learning.

6. Conclusions

In this paper, we propose CQEnsemble method based on cost-sensitive SVM and QBC to solve imbalanced data classification. CQEnsemble method divides the majority class into several subsets according to the proportion of imbalance samples. CQEnsemble method selects the effective training samples to join the last training set based on QBC active learning algorithm, so it avoids the shortages of the over-sampling and under-sampling. Experiment results show that the proposed method has higher *F-measure*, *G-mean,* and *AUC* than many existing class-imbalance learning methods.

Acknowledgments

This work is supported by China Postdoctoral Science Foundation (no. 20110491530) and Science Research Plan of Liaoning Education Bureau (no. L2011186).

References

[1] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[2] M. Gao, X. Hong, S. Chen, and C. J. Harris, "A combined SMOTE and PSO based RBF classifier for two-class imbalanced problems," *Neurocomputing*, vol. 74, pp. 3456–3466, 2011.

[3] G. Weiss, "Mining with rarity: a unifying framework," *SIGKDD Explorations*, vol. 6, no. 1, pp. 7–19, 2004.

[4] C. Seiffert, T. M. Khoshgoftaar, J. van Hulse, and A. Napolitano, "RUSBoost: a hybrid approach to alleviating class imbalance," *IEEE Transactions on Systems, Man, and Cybernetics A*, vol. 40, no. 1, pp. 185–197, 2010.

[5] M. S. Kim, "An effective under-sampling method for class imbalance data problem," in *Proceedings of the 8th Symposium on Advanced Intelligent Systems*, pp. 825–829, 2007.

[6] S. J. Yen and Y. S. Lee, "Cluster-based under-sampling approaches for imbalanced data distributions," *Expert Systems with Applications*, vol. 36, no. 3, pp. 5718–5727, 2009.

[7] X. Y. Liu, J. X. Wu, and Z. H. Zhou, "Exploratory undersampling for class-imbalance learning," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 39, no. 2, pp. 539–550, 2009.

[8] C. Drummond and R. C. Holte, "C4.5 decision tree, class imbalance, and cost sensitivity: why under-sampling beats over-sampling," in *Proceedings of the Workshop on Learning from Imbalanced Data Sets II, International Conference on Machine Learning*, 2003.

[9] N. V. Chawla, A. Lazarevic, L. O. Hall, and K. W. Bowyer, "SMOTEBoost: improving prediction of the minority class in boosting," in *Proceedings of the 7th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD '03)*, pp. 107–119, September 2003.

[10] S. Wang, Z. Li, W. Chao, and Q. Cao, "Applying adaptive over-sampling technique based on data density and cost-sensitive SVM to imbalanced learning," in *The International Joint Conference on Neural Networks (IJCNN '12)*, 2012.

[11] M. Gao, X. Hong, S. Chen, and C. J. Harris, "Probability density function estimation based over-sampling for imbalanced two-class problems," in *The International Joint Conference on Neural Networks (IJCNN '12)*, 2012.

[12] C. Elkan, "The foundations of cost-sensitive learning," in *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, pp. 973–978, 2001.

[13] B. X. Wang and N. Japkowicz, "Boosting support vector machines for imbalanced data sets," *Knowledge and Information Systems*, vol. 25, no. 1, pp. 1–20, 2010.

[14] Y. Sun, M. S. Kamel, A. K. C. Wong, and Y. Wang, "Cost-sensitive boosting for classification of imbalanced data," *Pattern Recognition*, vol. 40, no. 12, pp. 3358–3378, 2007.

[15] H. Guo and H. L. Viktor, "Learning from imbalanced data sets with boosting and data generation: the DataBoost-IM approach," *SIGKDD Explorations*, vol. 6, no. 1, pp. 30–39, 2004.

[16] R. Akbani, S. Kwek, and N. Japkowicz, "Applying support vector machines to imbalanced datasets," in *Proceedings of the 15th European Conference on Machine Learning (ECML '04)*, pp. 39–50, Pisa, Italy, September 2004.

[17] Y. Tang, Y. Q. Zhang, and N. V. Chawla, "SVMs modeling for highly imbalanced classification," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 39, no. 1, pp. 281–288, 2009.

[18] J. Wang, J. You, Q. Li, and Y. Xu, "Extract minimum positive and maximum negative features for imbalanced binary classification," *Pattern Recognition*, vol. 45, pp. 1136–1145, 2012.

[19] N. García-Pedrajas, J. Pérez-Rodríguez, and A. de Haro-García, "OligoIS: scalable instance selection for class-imbalanced data sets," *IEEE Transactions on Systems, Man, and Cybernetics B*, 2012.

[20] K. Veropoulos, C. Campbell, and N. Cristianini, "Controlling the sensitivity of support vector machines," in *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 55–60, 1999.

[21] H. S. Seung, M. Opper, and H. Sompolinsky, "Query by committee," in *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, pp. 287–294, July 1992.

[22] Y. Freund, H. S. Seung, E. Shamir, and N. Tishby, "Selective sampling using the query by committee algorithm," *Machine Learning*, vol. 28, no. 2-3, pp. 133–168, 1997.

[23] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," in *Proceedings of the 2nd European Conference on Computational Learning Theory*, pp. 23–37, 1995.

[24] M. V. Joshi, V. Kumar, and R. C. Agarwal, "Evaluating boosting algorithms to classify rare classes: comparison and improvements," in *Proceedings of the 1st IEEE International Conference on Data Mining (ICDM '01)*, pp. 257–264, December 2001.

[25] T. Fawcett, "ROC graphs: notes and practical considerations for researchers," Tech. Rep. HPL-2003-4, HP Labs, Palo Alto, Calif, USA, 2003.

[26] D. Lewis and W. Gale, "Training text classifiers by uncertainty sampling," in *Proceedings of the 7th Annual International ACM SIGIR Conference on Research and Development in Information*, pp. 73–79, New York, NY, USA, 1998.

[27] A. Frank and A. Asuncion, *UCI Machine Learning Repository*, University of California, School of Information and Computer Science, Irvine, Calif, USA, 2010, http://archive.ics.uci.edu/ml/.