Boosting Prediction Accuracy on Imbalanced Datasets with SVM Ensembles

Yang Liu, Aijun An, and Xiangji Huang

Department of Computer Science and Engineering,
York University,
Toronto, Ontario, M3J 1P3, Canada
{yliu, aan, jhuang}@cs.yorku.ca

Abstract. Learning from imbalanced datasets is inherently difficult due to lack of information about the minority class. In this paper, we study the performance of SVMs, which have gained great success in many real applications, in the imbalanced data context. Through empirical analysis, we show that SVMs suffer from biased decision boundaries, and that their prediction performance drops dramatically when the data is highly skewed. We propose to combine an integrated sampling technique with an ensemble of SVMs to improve the prediction performance. The integrated sampling technique combines both over-sampling and under-sampling techniques. Through empirical study, we show that our method outperforms individual SVMs as well as several other state-of-the-art classifiers.

1 Introduction

Many real-world datasets are imbalanced, in which most of the cases belong to a larger class and far fewer cases belong to a smaller, yet usually more interesting class. Examples of applications with such datasets include searching for oil spills in radar images [1], telephone fraudulent detection [2], credit card fraudulent detection diagnosis of rare diseases, and network intrusion detection. In such applications, the cost is high when a classifier misclassifies the small (positive) class instances.

Despite the importance of handling imbalanced datasets, most current classification systems tend to optimize the overall accuracy without considering the relative distribution of each class. As a result, these systems tend to misclassify minority class examples when the data is highly skewed. Techniques have been proposed to handle the problem. Approaches for addressing the problem can be divided into two main directions: sampling approaches and algorithm-based approaches. Generally, sampling approaches include methods that over-sample the minority class to match the size of the majority class [3, 4], and methods that under-sample the majority class to match the size of the minority class [1, 5, 6, 7]. Algorithmic-based approaches are designed to improve a classifier's performance based on their inherent characteristics.

This paper is concerned with improving the performance of the Support Vector Machines (SVMs) on imbalanced data sets. SVMs have gained success in

W.K. Ng, M. Kitsuregawa, and J. Li (Eds.): PAKDD 2006, LNAI 3918, pp. 107–118, 2006.

many applications, such as text mining and hand-writing recognition. However, when the data is highly imbalanced, the decision boundary obtained from the training data is biased toward the minority class. Most approaches proposed to address this problem have been algorithm-based [8, 9, 10], which attempt to adjust the decision boundary through modifying the decision function.

We take a complementary approach and study the use of sampling as well as ensemble techniques to improve SVM's performance. First, our observation indicates that using over-sampling alone as proposed in previous work (e.g. SMOTE [10]) can introduce excessive noise and lead to ambiguity along decision boundaries. We propose to integrate the two types of sampling strategies by starting with over-sampling the minority class to a moderate extent, followed by under-sampling the majority class to the similar size. This is to provide the learner with more robust training data. We show by empirical results that the proposed sampling approach outperforms over-sampling alone irrespective of the parameter selection. We further consider using an ensemble of SVMs to boost the performance. A collection of SVMs are trained individually on the processed data, and the final prediction is obtained by combining the results from those individual SVMs. In this way, more robust results can be obtained by reducing the randomness induced by a single classifier, as well as by alleviating the information loss due to sampling.

2 Related Work

Sampling is a popular strategy to handle the class imbalance problem since it straightforwardly re-balances the data at the data processing stage, and therefore can be employed with any classification algorithm [1, 3, 4, 5, 6, 7]. As one of the successful oversampling methods, the SMOTE algorithm [11] over-samples the minority class by generating interpolated data. It first searches for the K-nearest-neighbors for each minority instance, and for each neighbor, randomly selects a point from the line connecting the neighbor and the instance itself, which will serve as a new minority instance. By adding the "new" minority instances into training data, it is expected that the over-fitting problem can be alleviated. SMOTE has been reported to achieve favorable results in many classification algorithms [11, 12]. Algorithm-based approaches include methods in which existing learning algorithms are tailored to improve the performance for imbalanced datasets. For example, some algorithms consider class distributions or use cost functions for decision tree inductions [6, 13, 14].

SVMs have established themselves as a successful approach for various machine learning tasks. The class imbalance issue has also been addressed in the literature. Through empirical study, Wu et al. [9] report that when the data is highly imbalanced, the decision boundary determined by the training data is largely biased toward the minority class. As a result, the false negative rate that associates with the minority class might be high. To compensate for the skewness, they propose to enlarge the resolution around the decision boundary by revising kernel functions. Furthermore, Veropoulos et al. [8] use pre-specified penalty constants on Lagrange multipliers for different classes; Akbani et al. [10] combine SVMs with SMOTE over-sampling and cost sensitive learning. In

contrast, Japkowicz et al. [15] argue that SVMs are immune to the skewness of the data, because the classification decision boundary is determined only by a small quantity of support vectors. Consequently, the large volume of instances belonging to the majority class might be considered redundant. In this paper, we will demonstrate that the decision boundary changes as imbalance ratios vary, and discuss its implications.

Using an ensemble of classifiers to boost classification performance has also been reported to be effective in the context of imbalanced data. This strategy usually makes use of a collection of individually trained classifiers whose prediction results are integrated to make the final decision. The work in this direction includes that Chen et al. [6] use random forest to unite the results of decision trees induced from bootstrapping the training data, and that Guo et al [4] apply data boosting to improve the performance on hard examples that are difficult to classify. However, most current studies are confined to decision tree inductions instead of other classifiers, e.g, SVM. Moreover, decision-tree-based algorithms might be ill-suited for the class imbalance problem as they favor short trees.

3 Background

3.1 Support Vector Machines

In this section we briefly describe the basic concepts in two-class SVM classification. Assume that there is a collection of n training instances $Tr = \{x_i, y_i\}$, where $\mathbf{x_i} \in \mathcal{R}^N$ and $y_i \in \{-1, 1\}$ for $i = 1, \ldots, n$. Suppose that we can find some hyperplane which linearly separates the positive from negative examples in a feature space. The points $\mathbf{x}$ belonging to the hyperplane must satisfy $\mathbf{w} \cdot \mathbf{x} + b = 0$, where $\mathbf{w}$ is normal to the hyperplane and b is the intercept. To achieve this, given a kernel function K, a linear SVM searches for Lagrange multiplier α_i $(i = 1, ..., n)$ in Lagrangian

$$L_p \equiv \frac{1}{2}||w||^2 - \sum_{i=1}^{n} \alpha_i y_i (x_i \cdot w + b) + \sum_{i=1}^{n} \alpha_i \tag{1}$$

such that the margin between two classes $\frac{2}{||w||}$ is maximized in the feature space [16]. In addition, in the α_i optimizing process, Karush Kuhn Tucker (KKT) conditions which require $\sum_{i=1}^{n} \alpha_i y_i = 0$, must be satisfied.[1] To predict the class label for a new case x, we need to compute the sign of $f(x) = \sum_{i=1}^{n} y_i \alpha_i K(x, x_i) + b$. If the sign function is greater than zero, x belongs to the positive class, and the negative otherwise.

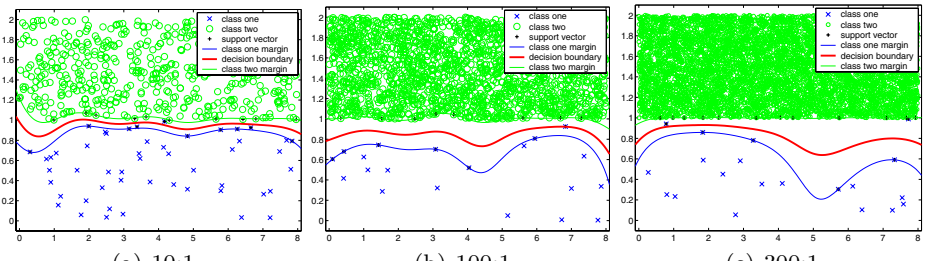
In SVMs, support vectors (SVs) are of crutial importance to the training set. They lie closest to the decision boundary; thus form the margin between

[1] In the case of non-separable data, 1-norm soft-margin SVMs minimize the Lagrangian $L_p = \frac{1}{2}||w||^2 + C\sum_i \xi_i - \sum_i \alpha_i \{y_i(x_i \cdot w + b) - 1 + \xi_i\} - \sum_i \mu_i \xi_i$, where ξ_i, $i \in [1, n]$ are positive slack variables, C is selected by users with a larger C indicating a higher penalty to errors, and μ_i are Lagrange multipliers to enforce ξ_i being positive. Similarly, corresponding KKT conditions have to be met for the purpose of optimization.

two sides. If all other training data were removed, and training was repeated, the same separating hyperplane would still be constructed. Note that there is a Lagrange multiplier α_i for each training instance. In this context, SVs correspond to those points for which $\alpha_i > 0$; other training instances have $\alpha_i = 0$. This fact gives us the advantage of classifying by learning with only a small number of SVs, as all we need to know is the position of the decision boundary which lies right in the middle of the margin; other training points can be considered redundant. Further, it is of prime interest in the class imbalance problem because SVMs could be less affected by the negative instances that lie far away from the decision boundary even if there are many of them.

3.2 Effects of Class Imbalance on SVMs

We conducted a series of experiments to investigate how the decision boundaries are affected by the imbalance ratio, i.e., the ratio between the number of negative examples and positive examples. We start with classifying a balanced training dataset, and detect that the real decision boundary is close to the "ideal boundary", as it is almost of equal length to both sides. We then reform successive new datasets with different degrees of data skewness by removing instances from the positive and add instances to the negative. Figure 1 reflects the data distribution when imbalance ratios vary from 10:1 to 300:1, where crosses and circles represent the instances from positive and negative classes respectively. From Figure 1 (a), we find that if the imbalance ratio is moderate, the boundary will still be close to the "ideal boundary". This observation demonstrates SVMs could be robust and self-adjusting; and is thus able to alleviate the problem arising from moderate imbalance. Nonetheless, as the imbalance ratio becomes larger and larger, as illustrated in Figure 1 (b) and (c), the boundaries get evidently biased toward the minority class. As a consequence, making predictions with such a system may lead to a high false negative rate.



(a) 10:1 (b) 100:1 (c) 300:1

Fig. 1. Boundary changes with different imbalance ratios

4 Re-balancing the Data

We have shown that SVMs may perform well while the imbalance ratio is moderate. Nonetheless, their performance could still suffer from the extreme data

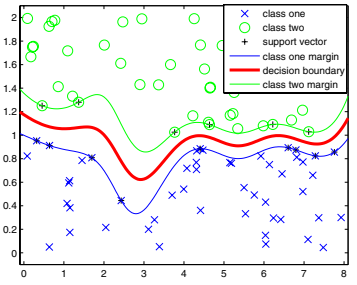


Fig. 2. Under-sampling majority instances

skewness. To cope with this problem, in this section, we study the use of sampling techniques to balance the data.

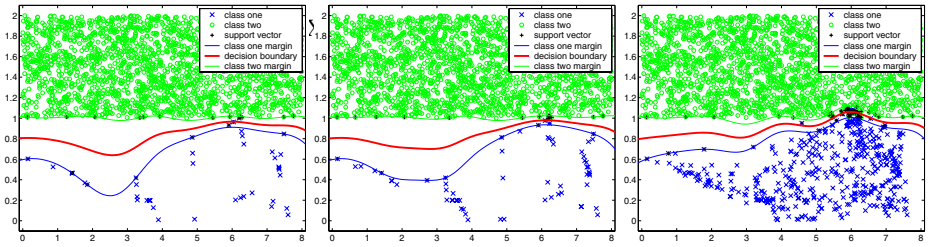
4.1 Undersampling

Under-sampling approaches have been reported to outperform over-sampling approaches in previous literatures. However, under-sampling throws away potentially useful information in the majority class; it thus could make the decision boundary trembling dramatically. For example, given the imbalance ratio as 100:1, in order to get a close match for the minority, it might be undesirable to throw away 99% of majority instances. Figure 2 illustrates such a scenario, where the majority class is undersampled to keep the same size as the minority, but a considerable amount of SVs lie far away from the ideal boundary $y = 1$. Accordingly, predicting with such SVMs may lead to low accuracies.

4.2 Oversampling

Considering that simply replicating the minority instances tends to induce overfitting, using interpolated data is often preferred in the hope of supplying additional and meaningful information on the positive class. SMOTE is the method that has been mostly cited along this line.

However, the improvement of integrating SVMs with the SMOTE algorithm can be limited due to its dependence on the proper selection of the number of nearest neighbors K as well as imbalance ratios. Basically, the value of K determines how many new data points will be added into the interpolated dataset. Figure 3 shows how the decision boundary will change with different K values. Figure 3 (a) shows the original class distribution while the imbalance ratio is 100:1. Figure 3 (b) demonstrates that the classification boundary is relatively smoothed when K has a small value; nonetheless, it is still biased toward the minority class. This is due to SMOTE actually providing little information of the minority; hence the oversampling in this case should be considered as a type of "phantom-transduction". When the interpolated dataset is considerably enlarged as K increases, as shown in Figure 3 (c), ambiguities could arise along the current boundary, because SMOTE makes the assumption that the instance between a positive class instance and its nearest neighbors is also positive. However



(a) ImbalanceRatio=100:1 (b) K=2 (c) K=19

Fig. 3. Using SMOTE with different K values

it may not be always true in practice. As a positive instance is very close to the boundary, its nearest neighbor is likely to be negative, and this possibility may increase as K and imbalance ratio become larger. Consequently, the new data instance, which actually belongs to the negative class, is mis-labeled as positive, and the induced decision boundary, as shown in Figure 3 (c), could be inversely distorted to the majority class.

4.3 Combination of Two Types of Samplings

To address the problems arising from using each of the two types of sampling approaches alone, we integrate them together. Given an imbalance ratio, we first over-sample the minority instances with SMOTE to some extent, and then under-sample the majority class so that both sides have the same or similar amount of instances. To under-sample the majority class, we use the bootstrap sampling approach with all available majority instances, provided that the size of the new majority class is the same as that of the minority class after running SMOTE. The benefit of doing so is that this approach inherits the strength of both strategies, and alleviates the over-fitting and information loss problems.

In addition, to avoid taking risks of inducing ambuities along the decision boundary, we choose to filter out the "impure" data firstly before sampling. In this context, an instance is defined to be "impure", if and only if two of its three nearest neighbors provide different class labels other than that of itself. This idea is motivated by the *Edited Nearest Neighbor Rule* [7], which was originally used to remove unwanted instances from the majority. In our work, however, to further reduce the uncertainty from both classes, such a filtering process is taken on each side.

5 Ensemble of SVMs

In this section, we present a method that uses an ensemble of SVM classifiers integrated with a re-balancing technique that combines both over-sampling and under-sampling. Re-balancing is still necessary in this context since in learning from extremely imbalanced data, it is very likely that a bootstrap sample used to train an SVM in the ensemble is composed of few or even none of the minority instances. Hence, each component learner of the ensemble would suffer from severe skewness, and the improvement of using an ensemble would be confined. Our proposed method, called *EnSVM*, is illustrated in Figure 4. As described in

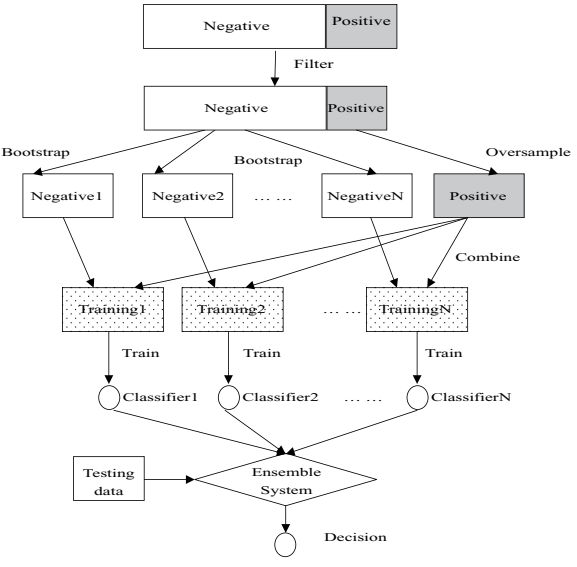


Fig. 4. EnSVM algorithm

Section 4.3, we start re-balancing the data by filtering out impurities which may induce ambiguities. Then, the minority class is over-sampled with the SMOTE method to smooth the decision boundary. That is, for each positive instance, it finds the K nearest neighbors, draws a line between the instance and each of its K nearest neighbors, and then randomly selects a point on each line to use as a new positive instance. In this way, $K \times n$ new positive instances are added to the training data, where n is the number of positive instances in the original training data. After that, we under-sample the majority class instances N times to generate N bootstrap samples so that each bootstrap sample has the same or similar size with the over-sampled positive instances. Then, each bootstrap sample (of the majority class) is combined with the over-sampled positive instances to form a training set to train an SVM. Therefore, N SVMs can be obtained from N different training sets. Finally, the N SVMs are combined to make a prediction on a test example by casting a *majority vote* from the ensemble of SVMs. In our experiments reported below, we set N to be 10.

6 Empirical Evaluation

In this section, we first introduce the evaluation measures used in our study, and then describe the datasets. After that, we report the experimental results that compare our proposed approach with other methods.

6.1 Evaluation Measures

The evaluation measures used in our experiments are based on the *Confusion Matrix*. Table 1 illustrates a confusion matrix for a two class problem with *pos-*

Table 1. Two-class confusion matrix

	Predicted Positive	*Predicted Negative*
Actual Positive	TP(True Positive)	FN(False Negative)
Actual Negative	FP(False Positive)	TN(True Negative)

itive and *negative* class values. With this matrix, our performance measures are expressed as follows:

- *g-mean* $= \sqrt{a^- \times a^+}$, where $a^- = \frac{TN}{TN+FP}$ and $a^+ = \frac{TP}{TP+FN}$;
- *F-measure* $= \frac{2 \times Precision \times Recall}{Precision+Recall}$, where $precision = \frac{TP}{TP+FP}$ and $recall = \frac{TP}{TP+FN}$.

G-mean is based on the recalls on both classes. The benefit of selecting this metric is that it can measure how balanced the combination scheme is. If a classifier is highly biased toward one class (such as the majority class), the *g-mean* value is low. For example, if $a^+ = 0$ and $a^- = 1$, which means none of the positive examples is identified, *g-mean*=0. In addition, *F-measure* combines the recall and precision on the positive class. It measures the overall performance on the minority class. Besides, we utilize the ROC analysis [17] to assist the evaluation. A ROC curve demonstrates a trade off between true positive and false positive rates provided with different classification parameters. Informally, one point in ROC space is superior to another if it is closer to the northwest corner (TP is higher, but FP is lower). Thus, ROC curves allow for a visual comparison of classifiers: the larger the area below the ROC curve, the higher classification potential of the classifier.

6.2 Benchmark Data

We use five datasets as our testbeds. Four of the datasets are from the UCI Machine Learning Repository and another dataset is a medical compound dataset (mcd) collected by National Cancer Institute (NCI) for discovering new compounds capable of inhibiting the HIV virus. The four UCI datasets are *spambase, letter-recognition, pima-indians-diabetes* and *abalone*. Each dataset in this study is randomly split into training and test subsets of the same size, where a stratified manner is employed to ensure that the training and test sets have the same imbalance ratio. Table 2 shows the characteristics of the five datasets. The first

Table 2. Benchmark datasets

Dataset	*Datapoints*	*Attributes*	*ImbalanceRatio*
letter	20000	16	2:1
pima	768	9	2:1
spambase	3068	57	10:1
abalone	4280	8	40:1
mcd	29508	6	100:1

three datasets (letter, pima, and spambase) are mildly imbalanced, while the next two (abalone and mcd) are very imbalanced. These datasets were carefully selected to (1) fulfill the requirements that they are obtained in real applications, (2) distinct from feature characteristics, and vary in size and imbalance ratio, and (3) maintain sufficient amount of instances in each individual class to keep the classification performance.

6.3 Experimental Results

In this section, we compare the performance of our proposed *EnSVM* method with those of five other methods: 1) single SVM without re-sampling the data, 2) single SVM with over-sampling using SMOTE [10] (without applying cost functions), 3) random forest with balanced training data from under-sampling [6], 4) random forest with our combined sampling method, and 5) single SVM with our combined sampling method. In our experiments, for all the SVMs, we employed Gaussian RBF kernels of the form $K(x_i, x_j) = \exp(-\gamma|x_i - x_j|^2)$ of C-SVMs. For each method we repeated our experiments ten times, computed average g-mean values and F-measures.

Table 3. Performance in terms of g-mean

Dataset	*SVM*	*SMOTE K=1*	*SMOTE K=highest*	*RandForest*[1]	*RandForest*[2]	*AvgSVM*	*EnSVM K=1*	*EnSVM K=highest*
letter	0.9551	0.9552	0.9552	0.9121	0.9281	0.9563	**0.9566**	**0.9566**
pima	0.6119	0.7320	0.7320	0.7358	0.7002	0.7419	**0.7503**	**0.7503**
spam	0.8303	0.8364	0.8580	0.8593	**0.9050**	0.8592	0.8616	0.8988
abalone	0.6423	0.6280	0.8094	0.7358	0.7678	0.8041	**0.8958**	0.8311
mcd	0.4500	0.4496	0.5952	0.5896	0.5968	0.5931	0.5951	**0.6039**

Results in terms of g-mean are shown in Table 3, where *SVM* denotes the single SVM method with the original training data, *SMOTE* represents oversampling the minority class and then training a system with single SVMs, *RandForest*[1] denotes undersampling the majority class and then making an ensemble with C4.5 decision trees, *RandForest*[2] denotes sampling data with our combined method, followed by forming an ensemble with C4.5, *AvgSVM* denotes the average performance of 10 single SVMs with our sampling method, and *EnSVM* is our ensemble method with the combined sampling method. For the first two datasets, the K values for *SMOTE* and *EnSVM* can only be set to be 1 since their imbalance ratio is 2:1. For each of other datasets, we test two K values: the smallest value, which always equals to 1, and the highest value. The latter will depend on the imbalance ratios of three datasets, which are 9, 39, and 99 respectively. From the results we can see that *EnSVM* achieves the best results on all the datasets except on the spam dataset for which *RandForest*[2] is the best. [2]

Table 4 shows the performance for each method in terms of F-measure. We find that *EnSVM* deserves the highest value on all five datasets. In particular, a big improvement is made on the datasets where the imbalance ratios are large. By comparing the results from the four SVM methods, we can see that (1) using SMOTE to over-sample the data is better than SVM without sampling; (2) using our combined sampling method with single SVMs is better than using only over-sampling with SMOTE; and (3) using the ensemble method together with the combined sampling method achieve the best results. By comparing the two Random Forest methods, using the combined sampling method is better than

[2] In Table 3, from top to bottom, the optimal γ obtained empirically in using SVMs is 1.0×10^{-2}, 5.0×10^{-5}, 7.0×10^{2}, and 10^{2} respectively. In addition, C is set to be 1000 for each case.

Table 4. Performance in F-measure

Dataset	*SVM*	*SMOTE K=1*	*SMOTE K=highest*	*RandForest*[1]	*RandForest*[2]	*AvgSVM*	*EnSVM K=1*	*EnSVM K=highest*
letter	0.9548	0.9549	0.9549	0.9111	0.9268	0.9406	**0.9563**	**0.9563**
pima	0.5664	0.7135	0.7135	0.7098	0.6165	0.7259	**0.7357**	**0.7357**
spam	0.8164	0.8238	0.8492	0.8512	0.8751	0.7498	0.8553	**0.8950**
abalone	0.5843	0.5659	0.7938	0.7938	0.7426	0.7875	**0.8940**	0.8190
mcd	0.3367	0.3364	0.5285	0.5285	0.5286	0.5274	0.5272	**0.5415**

using only the under-sampling method on most datasets. Moreover, between the Random Forest method and the ensemble of SVMs method, the latter performs better.

In addition to the imbalance ratio, the selection of K may also impact on the prediction accuracy of *SMOTE* and *EnSVM*. To make a better understanding, we present a *ROC* analysis result with the *spambase* dataset. This dataset is considered since it has a moderate imbalance ratio and instance volume. The original *spambase* has an imbalance ratio of 10; therefore, in this experiment, we test K from 1 to 9, and depict the ROC curves of the two approaches in Figure 5. Clearly, compared to simply over-sampling the minority instances, *EnSVM* generates a better result. We also test how the g-mean value may change with different Ks in *SMOTE* and *EnSVM*. The *abalone* and *mcd* datasets are used in this case as they hold large imbalance ratios and allow K to vary in relatively large ranges. We set parameter K to vary from 1 to 39 for the *abalone* dataset and from 1 to 99 for the *mcd* dataset. As shown in Figures 6.3 (a) and (b), the prediction performance of *EnSVM* is superior to simply applying the *SMOTE* algorithm with respect to each K value. Moreover, we can see that the optimal K value can be difficult to determine in both *SMOTE* and *EnSVM*. For *EnSVM*, when K is small, we get *better* neighbors for the oversampling process, so the prediction performance can be dramatically improved. Further, when K is big, more noise is likely to be introduced, but a larger training data set is generated using *EnSVM* and less information is lost. Consequently, it becomes a trade off between inducing more noise and losing less information. Nonetheless, our method is better than *SMOTE* with all K values.

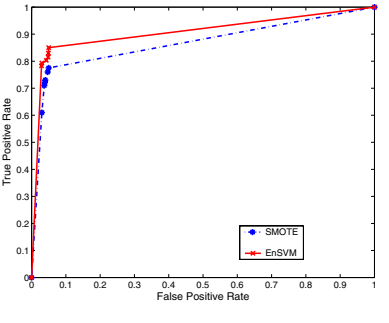
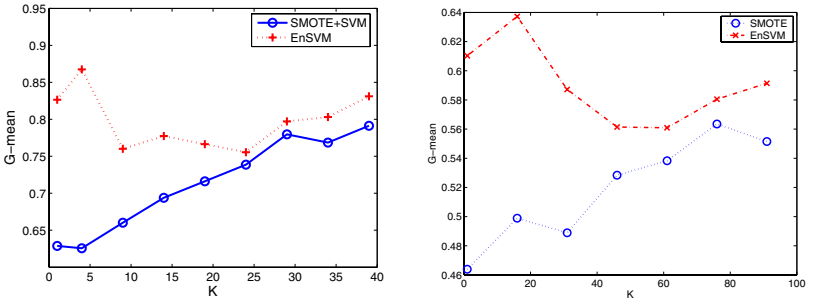


Fig. 5. ROC curve of spambase dataset



(a) abalone (b) mcd

Fig. 6. G-mean wrt. different K values

7 Conclusions

This paper introduces a new approach to learning from imbalanced datasets through making an ensemble of SVM classifiers and combining both over-sampling and under-sampling techniques. We first show in this study that using SVMs for class prediction can be influenced by the data imbalance, although SVMs can adjust itself well to some degree of data imbalance. To cope with the problem, re-balancing the data is a promising direction, but both undersampling and oversampling have limitations. In our approach, we integrate the two types of sampling strategies together. Over-sampling the minority class provides complementary knowledge for the training data, and under-sampling alleviates over-fitting problem. In addition, we make an ensemble of SVMs to enhance the prediction performance by casting a majority vote. Through extensive experiments with real application data, our proposed method is shown to be effective and better than several other methods with different data sampling methods or different ensemble methods. We are now working on a method for automatically determining the value of K based on the data set characteristics in order to optimize the performance of *EnSVM*.

References

1. Kubat, M., Holte, R.C., Matwin, S.: Machine learning for the detection of oil spills in satellite radar images. Mach. Learn. **30** (1998) 195–215
2. Fawcett, T., Provost, F.J.: Adaptive fraud detection. Data Mining and Knowledge Discovery **1** (1997) 291–316
3. Ling, C.X., Li, C.: Data mining for direct marketing: Problems and solutions. In: KDD. (1998) 73–79
4. Guo, H., Viktor, H.L.: Learning from imbalanced data sets with boosting and data generation: the databoost-im approach. SIGKDD Explorations **6** (2004) 30–39
5. Kubat, M., Matwin, S.: Addressing the curse of imbalanced training sets: one-sided selection. In: Proc. 14th International Conference on Machine Learning. (1997) 179–186

6. Chen, C., Liaw, A., Breiman, L.: Using random forest to learn imbalanced data. Technical Report 666, Statistics Department, University of California at Berkeley (2004)
7. Wilson, D.R., Martinez, T.R.: Reduction techniques for instance-basedlearning algorithms. Mach. Learn. **38** (2000) 257–286
8. Veropoulos, K., Cristianini, N., Campbell, C.: Controlling the sensitivity of support vector machines. In: International Joint Conference on Artificial Intelligence(IJCAI99). (1999)
9. Wu, G., Chang, E.Y.: Aligning boundary in kernel space for learning imbalanced dataset. In: ICDM. (2004) 265–272
10. Akbani, R., Kwek, S., Japkowicz, N.: Applying support vector machines to imbalanced datasets. In: ECML. (2004) 39–50
11. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: Synthetic minority over-sampling technique. J. Artif. Intell. Res. (JAIR) **16** (2002) 321–357
12. Chawla, N.V., Lazarevic, A., Hall, L.O., Bowyer, K.W.: Smoteboost: Improving prediction of the minority class in boosting. In: PKDD. (2003) 107–119
13. Weiss, G.M., Provost, F.J.: Learning when training data are costly: The effect of class distribution on tree induction. J. Artif. Intell. Res. (JAIR) **19** (2003) 315–354
14. Drummond, C., Holte, R.C.: C4.5, class imbalance, and cost sensitivity: Why under-sampling beats over-sampling. In: Workshop on Learning from Imbalanced Datasets II held in conjunction with ICML'2003. (2003)
15. Japkowicz, N., Stephen, S.: The class imbalance problem: A systematic study. Intell. Data Anal. **6** (2002) 429–449
16. Burges, C.J.C.: A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery **2** (1998) 121–167
17. Swets, J.: Measuring the accuracy of diagnostic systems. Science **240** (1988) 1285–1293