Adaptive Sampling with Optimal Cost for Class-Imbalance Learning

Yuxin Peng
Institute of Computer Science and Technology, Peking University, Beijing 100871, China
pengyuxin@pku.edu.cn

Abstract

Learning from imbalanced data sets is one of the challenging problems in machine learning, which means the number of negative examples is far more than that of positive examples. The main problems of existing methods are: (1) The degree of re-sampling, a key factor greatly affecting performance, needs to be pre-fixed, which is difficult to make the optimal choice; (2) Many useful negative samples are discarded in under-sampling; (3) The effectiveness of algorithm-level methods are limited because they just use the original training data for single classifier. To address the above issues, a novel approach of adaptive sampling with optimal cost is proposed for class-imbalance learning in this paper. The novelty of the proposed approach mainly lies in: adaptively over-sampling the minority positive examples and under-sampling the majority negative examples, forming different sub-classifiers by different subsets of training data with the best cost ratio adaptively chosen, and combining these sub-classifiers according to their accuracy to create a strong classifier. It aims to make full use of the whole training data and improve the performance of class-imbalance learning classifier. The solid experiments are conducted to compare the performance between the proposed approach and 12 state-of-the-art methods on challenging 16 UCI data sets on 3 evaluation metrics, and the results show the proposed approach can achieve superior performance in class-imbalance learning.

Introduction

Class-imbalance learning issue often occurs in classification, which means the number of negative examples is far more than that of positive examples in the training data, and is popular in practice. The class-imbalance learning is a major factor to affect the performance of classifiers, which has been extensively studied in the existing works. Generally speaking, the existing methods can be divided into two categories: data-level approaches and algorithm-level approaches. The data-level approaches re-sample the data set to get a more balanced class distribution, which includes under-sampling and over-sampling methods. And the algorithm-level approaches do not change the data set, but try to make the minority samples more important than the majority samples by adjusting the cost of samples.

In the data-level approaches, the random over-sampling of minority positive samples and the random under-sampling of majority negative samples are two concise but effective methods (Batista, Prati, & Monard 2004; Estabrooks, Jo, & Japkowicz 2004; Molinara, Ricamato, & Tortorella 2007). More complex over-sampling methods include the synthetic minority over-sampling technique (SMOTE) (Chawla, Hall, & Bowyer 2002) which creates new "synthetic" samples between the existing minority samples, the borderline-SMOTE algorithm (Han, Wang, & Mao 2005) which is a modification of SMOTE, the adaptive synthetic sampling approach (ADASYN) (He et al. 2008) which generates more synthetic minority class samples from hard samples with a weighted distribution for minority class examples, and the DataBoost-IM (Guo & Viktor 2004) approach which combines boosting, ensemble-based learning algorithm and data generation to solve the class imbalance problem. Liu, Wu, & Zhou (2006, 2009) propose two under-sampling approaches named EasyEnsemble and BalanceCascade to study the class-imbalance problem, and the experiment result showed the effectiveness of the two approaches. Ertekin, Huang, & Giles (2007) adopt the active learning algorithm. And Yuan, Li, & Zhang (2006) propose the support cluster machines (SCMs) algorithm to under-sample the majority class through an iteration of the clustering step and shrinking step. In addition, Tang et al. (2009) propose granular SVMs-repetitive under-sampling algorithm (GSVM-RU), which utilizes SVMs to select the negative samples called negative local support vectors.

In the algorithm-level approaches, some assign different costs to the samples of different classes, and make the minority positive samples more important than the majority negative samples during the training process. They adopt the different penalty constants C+ and C− for the positive and negative examples, which have been reported to be effective (Sun, Kamel, & Wang 2006; Nguyen, Zeno, & Lars 2010). However, Wu & Chang (2005) point out that

Copyright © 2015, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

the effectiveness of this method is limited. In addition, some complex algorithm-level methods have also been presented. For example, Wu & Chang (2003) present an adaptive conformal transformation (ACT) algorithm to change the kernel function of SVM. Chen, Lu, & Kwok (2006) address the class-imbalance problem with min-max modular network. And Cao, Zhao, & Zaiane (2013) propose an optimized cost-sensitive SVM incorporating two evaluation measures (AUC and G-mean).

In summary, the main problems of the existing methods are: (1) In most existing data-level methods, the degree of re-sampling, which is a key factor that affects greatly the performance (Chawla et al. 2002; Han, Wang, and Mao 2005), needs to be pre-fixed as a parameter by the user (e.g., "generate 3 times synthetic samples as the original positive samples" (Batista, Prati, and Monard 2004)), or decided by some heuristic rules (e.g., "after re-sampling, positive and negative sets should be of comparable or equal size" (Liu, Wu, & Zhou 2006, 2009)). However, the pre-fixed parameters or heuristic rules are not generally the optimal choice for the degree of re-sampling; (2) Many useful negative samples are discarded in most under-sampling methods (Ertekin et al. 2007; Yuan, Li, & Zhang 2006); (3) The effectiveness of cost-sensitive learning methods is limited, since the Karush Kuhn Tucker (KKT) conditions in SVM algorithm impose the influence from positive and negative support vectors equally to classification (Wu & Chang 2005).

To address the above issues, a novel approach of adaptive sampling with optimal cost is proposed for class-imbalance learning in this paper. The novelty of the proposed approach is as follows: instead of fixing the sampling degree, the proposed approach adaptively over-sample the positive examples and under-sample the negative examples to form different sampled datasets, train sub-classifiers on these sampled datasets by the proposed adaptive cost-sensitive learning method, and combine these sub-classifiers according to their accuracy to create a strong classifier. It aims to make full use of the whole training data and improve the performance of the class-imbalance learning classifier. In the proposed approach, firstly, the majority negative examples are divided into some disjoint subsets by the weighting-based under-sampling method. Then, for each subset of negative examples, the SMOTE algorithm is utilized to over-sample the positive examples with different size based on the proposed weighting distribution function. Sub-classifiers are trained using each subset with the best cost ratio chosen adaptively, and are fused with different weights to get a weak classifier. Finally, these classifiers from each subset of negative examples are combined to create a strong classifier. The solid experiments are conducted to compare the performance between the proposed approach and 12 state-of-the-art methods on challenging 16 UCI data sets on 3 evaluation metrics, and the results show the proposed approach can achieve superior performance in class-imbalance learning.

The proposed Approach

Algorithm Framework

Given the training data set $\{(x_1, y_1), (x_2, y_2), \ldots, (x_k, y_k)\}$, where y_i=-1,+1 for the negative and positive examples respectively. Denote the set of positive samples as P and the set of negative samples as N. The framework of the proposed approach is presented as follows, and the details will be shown later.

Step 1: Weighting-Based Sampling

A weight value is calculated for each sample, and the negative samples are divided into m disjoint subsets $N_1, N_2, \ldots, N_m$, i.e. $N_1+N_2+\ldots+N_m=N$, where "+" denotes the union operation of set. Subsequently, the positive set P is over-sampled to $2P, 3P, \ldots, (n+1)P$ by my modification of the borderline-SMOTE algorithm based on the weights, where jP denote the set with P and $(j-1)P$ new synthetic positive samples. The m and n is defined as follows.

$$m = \sqrt{2R / \log R} \tag{1}$$

$$n = \sqrt{R / 2\log R} \tag{2}$$

where R is the ratio of negative samples versus positive samples. Formula (1) and (2) are derived by the heuristic rules, and $m=2n$ is because under-sampling is generally more effective than over-sampling.

Step 2: Adaptive Cost-Sensitive Learning

For each negative subset N_i and positive set $(j+1)P$ as training set, a $sub\text{-}classifier_{ij}$ with N_i and $(j+1)P$ is constructed by the adaptive cost-sensitive learning method, which adaptively select the best cost ratio for classifier.

Step 3: Fusion of Classifiers

After constructing the sub-classifiers, m weak-classifiers $classifier_i$ are constructed by weighted fusion of the sub-classifiers for each N_i, and the final strong-classifier is constructed by weighted fusion of the m weak-classifiers.

Weighting-Based Sampling

In above step 1, the weighting-based sampling is composed of three parts: weight calculation, weighting-based under-sampling and weighting-based over-sampling, which are shown as follows. Moreover, the details of the modification of borderline-SMOTE are also presented.

Part 1: Weight Calculation

In the weighting-based sampling method, the weight for each negative example and positive example is calculated before sampling. Then, the data set is sampled based on the weight, which is set according to the probabilistic value of

classifier output. In a classification model which outputs the probability of a sample to be positive, the higher probability a negative sample gets, the more important this sample is, since this negative sample is more likely to be misclassified to be a positive sample, and may contain more useful information for classification. Correspondingly, the lower probability a positive sample gets, the more important this sample is. The weight of each example is calculated as described in Algorithm 1.

Algorithm 1: Calculate the weight of training data

Input:
The original training data set labeled as positive set P and negative set N.
Output:
The weight function $Wp(p_i)$ for positive set P and $Wn(n_i)$ for negative set N.
repeat
1. Randomly divide P into P_1, P_2 and N into N_1, N_2 *s.t.* $|P_1|=|P_2|$ and $|N_1|=|N_2|$
2. $model_1 \leftarrow SVMTrain(P_1+N_1)$
   $model_2 \leftarrow SVMTrain(P_2+N_2)$
3. $i \leftarrow 0$
   repeat
   $i \leftarrow i+1$
   if $(x_i, y_i) \in P_1+N_1$
   $prob(x_i) \leftarrow SVMPredict(model_2, x_i)$
   else
   $prob(x_i) \leftarrow SVMPredict(model_1, x_i)$
   end if
   until $i=|P|+|N|$
4. Calculate the weight in iteration t:

$$Wp_t(p_i) \leftarrow 1 - prob(p_i), p_i \in P$$
$$Wn_t(n_i) \leftarrow prob(n_i), n_i \in N$$

until criteria is met
Calculate the final weight of each example:

$$Wp(p_i) \leftarrow \sum Wp_t(p_i) / Z_p$$
$$Wn(n_i) \leftarrow \sum Wn_t(n_i) / Z_n$$

where $Z_p = \sum Wp(p_i)$ and $Z_n = \sum Wn(n_i)$.

Part 2: Weighting-Based Under-Sampling

After the weight is calculated, the weighting-based under-sampling method is proposed as the follows.

1) Sort the negative samples N by the descending order of weight. The sorted negative samples are listed as $\{n_1, n_2, \ldots, n_{|N|}\}$.

2) Divide N into $|N|/m$ disjoint subsets $N_1, N_2, \ldots, N_{|N|/m}$ with equal size:

$$N_i = \{n_{(i-1)m+1}, n_{(i-1)m+2}, \ldots, n_{im}\} \quad (3)$$

3) Divide randomly $N_i (i=1,2, \ldots, |N|/m)$ into m disjoint subsets $N_{i1}, N_{i2}, \ldots, N_{im}$ with equal size, that is:

$$N_i = \bigcup_{j=1}^{m} N_{ij}, i = 1, 2, \ldots, |N|/m \quad (4)$$

4) N_i' is obtained by selecting the subsets $N_{1i}, N_{2i}, \ldots, N_{|N|/m\,i}$ from N:

$$N_i' = \bigcup_{j=1}^{|N|/m} N_{ji}, i = 1, 2, \ldots, m \quad (5)$$

In this way, the set of negative examples N is split into m disjoint subsets $N_1', N_2', \ldots, N_m'$ with the same size.

The advantage of the weighting-based under-sampling is: 1) It can keep the diversity of negative samples for the subsets of N, because N_i' is obtained from all distribution of N, and the distribution of N_i' and N are nearly the same, while the random-based under-sampling cannot keep the diversity and the distribution. 2) The whole negative data set is used in the training process, so the whole information of negative examples can be used for classification.

Part 3: Weighting-Based Over-Sampling

The weighting-based over-sampling method is proposed for $sub\text{-}classifier_{ij}$ as the follows.

1) For each positive example p_i in P, calculate np_i as the number of samples that needs to be generated from p_i:

$$np_i = Wp(p_i) \times j \times |P| \quad (6)$$

where j is over-sampling rate of $sub\text{-}classifier_{ij}$.

2) For each p_i, generate np_i new synthetic positive samples with the modification of Borderline-SMOTE approach and construct the over-sampled synthetic positive data set with the size of $(j+1)|P|$.

The advantage of weighting-based over-sampling is: It can generate more new synthetic samples from "important" samples near the boundary. The synthetic examples better balance the number of examples near the boundary and alleviate the bias of learning, and are more useful in classification.

Modification of Borderline-SMOTE

In the sampling step, the proposed algorithm adopts a modification of borderline-SMOTE (Han, Wang, & Mao 2005), to generate new synthetic positive samples as follows: For each positive sample p, calculate its Euclidean distances with all samples in $P+N_i$. Find the m nearest neighboring samples with p, and denote the number of negative samples in them as m'. The new synthetic positive samples are generated differently in the following two cases.

1) If $m'=m$, that is, if all of the m nearest samples are negative samples, randomly select a negative sample n_k in the m nearest samples, and generate a new synthetic positive sample as $s_k=p+(n_k-p)r_k$, where r_k is a random number and $0<r_k<0.5$.

2) If $m'<m$, randomly select a positive sample p_k in the m nearest samples, and generate a new synthetic positive sample as $s_k=p+(p_k-p)r_k$, where r_k is a random number and $0<r_k<0.5$.

Adaptive Cost-Sensitive Learning

After re-sampling the data set, the adaptive cost-sensitive learning method is proposed to train classifiers. For each $sub\text{-}classifier_{ij}$, which uses $(j+1)P+N_i$ as training data set, the "best" cost ratio $C+/C-$ of the penalty constants $C+$ and $C-$ is adaptively selected for positive and negative samples, so the "best" model for $sub\text{-}classifier_{ij}$ is constructed. T

cost values are calculated equally-spaced in the candidate range (1 to *MaxCost*), and a classifier is trained with each cost value. The best cost ratio is selected which achieves the highest accuracy. The detailed algorithm is presented in the Algorithm 2.

Algorithm 2: Select the BestCostRatio

Input:
Training set $(j+1)P$ and N_i
Testing set $P+(N-N_i)$
Output:
BestCostRatio
Initialize:
$BestAccuracy \leftarrow 0$
$ImbRatio \leftarrow |N_i| / (j+1)|P|$
$MaxCost \leftarrow 2 \times ImbRatio$
$k \leftarrow 0$
repeat
1. $k \leftarrow k+1$
2. $CurrCost \leftarrow 1+(MaxCost-1)/(T-1)\times(k-1)$
3. $model_k \leftarrow SVMTrain((j+1)P+ N_i, CurrCost)$
4. $accuracy_k \leftarrow GetAccuracy(model_k, P+(N-N_i))$
5. **if** $accuracy_k > BestAccuracy$ **then**
   $BestAccuracy \leftarrow accuracy_k$
   $BestCostRatio \leftarrow CurrCost$
   end if

until $k=T$

Fusion of Classifiers

In the previous steps, $m \times n$ sub-classifiers are constructed. For each negative subset N_i, the weak-classifier $classifier_i$ is constructed as the weighted fusion of n sub-classifiers constructed from N_i with $2P$, $3P$,…, $(n+1)P$. The sub-classifiers constructed from N_i are applied to predict the set $P+(N-N_i)$ and get the *F-measure* as accuracy as follows.

$$F\text{-}measure = 2 \times Precision \times Recall / \left(Precision + Recall \right) \quad (7)$$

$$Precision = TP / \left(TP + FP \right) \quad (8)$$

$$Recall = TP / \left(TP + FN \right) \quad (9)$$

where *TP* denote the number of positive samples which is correctly classified, while *FP* and *FN* denote the number of misclassified positive and negative examples respectively, as shown in Table 1.

Table 1: Confusion matrix for a two-class problem.

	Predicted positive	Predicted negative
Positive	*TP* (true positive)	*FN* (false negative)
Negative	*FP* (false positive)	*TN* (true negative)

The weak-classifier $classifier_i$ is weighted fusion of n sub-classifiers and the weights are set according to the accuracy as follows.

$$classifier_i = \sum_{j=1}^{n} weight_{ij} \times sub\text{-}classifier_{ij} \quad (10)$$

$$weight_{ij} = accuracy_{ij} / \sum_{j=1}^{n} accuracy_{ij} \quad (11)$$

The final strong-classifier is defined as the weighted fusion of the above m weak-classifiers, the weights of which are set according to their accuracy evaluated on validation set $P+N'$, where N' is generated by selecting randomly a certain number of samples from each N_i defined in the sampling step. The weighted fusion for final classifier is defined as follows.

$$classifier = \sum_{i=1}^{m} weight_i \times classifier_i \quad (12)$$

$$weight_i = accuracy_i / \sum_{i=1}^{m} accuracy_i \quad (13)$$

The average fusion of the m classifiers is adopted based on the subsets of negative samples to get the final strong classifier as (14), because predicting the $P+N'$ to evaluate the weak-classifiers is time consuming, and the differences are small among the weak-classifiers. In addition, the average fusion can also improve the computational speed. The experimental results show this algorithm with average fusion of weak-classifiers is both efficient yet effective.

$$classifier = \frac{1}{m} \sum_{i=1}^{m} classifier_i \quad (14)$$

Experiments

Evaluation Metrics

In the experiments, three metrics, namely *AUC*, *F-measure* and *G-mean*, are jointly adopted to comprehensively evaluate the performance of the proposed approach, which is a very strict evaluation. In all three metrics, the higher score the classifier achieves, the better the performance is. *F-measure* is defined as (7), and *G-mean* is defined as follows, where the meaning of *TP*, *FP*, *TN*, and *FN* are described in Table 1.

$$G-mean = \sqrt{Recall \times Specificity} \quad (15)$$

$$Specificity = TN / (TN + FP) \quad (16)$$

The *area under the ROC curve* (*AUC*) has been proved to be a reliable evaluation metric for classification on imbalanced data set (Folleco, Khoshgoftaar, & Napolitano 2008). The receiving operating characteristic (*ROC*) curve depicts true positive rate versus the false positive rate. *ROC* curves illustrate the performance across all possible value of false positive rate.

Data Set Description

In the experiments, the 16 UCI data sets (Frank, & Asuncion 1998) are adopted to evaluate the proposed approach. These data sets come from different areas of life, with significant diversity, which is summarized in Table 2.

Experiment Setup

For each data set, a ten-fold stratified cross validation is performed, and for each fold, the classification is repeated for ten times to reduce the influence of randomness. The

Table 2: Information of 16 UCI datasets.

Dataset	Size	Attribute	Maj/Min	Ratio
car	1,728	8	1344/384	3.5
ionosphere	351	4	225/126	1.8
letter	20,000	6	19211/789	24.3
phoneme	5,404	9	3818/1586	2.4
satimage	6,435	3	5809/626	9.3
wdbc	569	13	357/212	1.7
abalone	4,177	33	3786/391	9.7
balance	625	16	576/49	11.8
cmc	1,473	6	1140/333	3.4
haberman	306	47	225/81	2.8
housing	506	5	400/106	3.8
mf-morph	2,000	8	1800/200	9.0
mf-zernike	2,000	36	1800/200	9.0
pima	768	18	500/268	1.9
vehicle	846	30	634/212	3.0
wpbc	198	33	151/47	3.2

whole cross validation process is repeated for five times to avoid any bias that may occur in random selection. The *AUC*, *F-measure*, and *G-mean* are averaged from all of the runs for comparison. The entire process of experiments is exactly the same as (Liu, Wu, & Zhou 2006, 2009), so that the results can be fairly compared with the Balance-Cascade and Easy-Ensemble methods proposed by (Liu, Wu, & Zhou 2006, 2009). The proposed approach is also compared to 10 other methods, which are experimentally compared in (Liu, Wu, & Zhou 2006, 2009). In addition, the basic SVM classifier is applied without considering the class-imbalance learning as baseline for comparison, that is, directly use the original positive and negative examples as training set. Totally, 14 methods are compared, which are presented as follows.

1. SVM.
2. Bagging (abbreviated as Bagg) (Breiman 2001).
3. AdaBoost (abbreviated as Ada) (Schapire 1999).
4. AsymBoost (abbreviated as Asym) (Viola & Jones 2002).
5. Under-Sampling + AdaBoost (abbreviated as Under) (Liu, Wu, & Zhou 2006, 2009).
6. SMOTE+AdaBoost (abbreviated as SMOTE) (Chawla et al. 2002).
7. Chan and Stolfo's method + AdaBoost (abbreviated as Chan) (Chan & Stolfo 1998).
8. Random Forests (abbreviated as RF) (Breiman 1996).
9. Balanced Random Forests (abbreviated as BRF) (Chen, Liaw, & Breiman 2004).
10. Under-Sampling + Random Forests (abbreviated as Under-RF) (Liu, Wu, & Zhou 2006, 2009).
11. Over-Sampling + Random Forests (abbreviated as Over-RF) (Liu, Wu, & Zhou 2006, 2009).
12. Balance-Cascade (abbreviated as Cascade) (Liu, Wu, & Zhou 2006, 2009).
13. Easy-Ensemble (abbreviated as Easy) (Liu, Wu, & Zhou 2006, 2009).
14. The proposed approach.

In all above methods, for each subset of the training data, SVM is used to train a sub-classifier with the *LibSVM* implementation, *RBF* kernel and default parameters.

Experiment Result

The results of 14 methods are shown in Table 3, 4, and 5 on *AUC*, *F-measure* and *G-mean* respectively. The best result of each data set is indicated with bold font and underline. The following conclusions can be obtained:

On all three evaluation metrics, the proposed approach stably achieves the best results among all 14 methods. The proposed approach achieves the highest scores of 0.884 on AUC, 0.658 on F-measure, and 0.820 on G-mean. On *AUC*, *F-measure*, and *G-mean*, the proposed approach is respectively 2.6%, 3.0%, and 1.4% higher than the best results of 13 methods on average. However, the score of (Liu, Wu, & Zhou 2006, 2009) is only 0.3%, 0.9%, and 1.4% higher than the best results of other 12 methods on average, which indicates the great difficulty on improving the accuracy on all 3 evaluation metrics of 16 data sets. Furthermore, the proposed approach achieves the best result on 13 data sets on *AUC*, on 12 data sets on *F-measure*, and on 11 data sets on *G-mean*. On the rest data sets, the proposed approach also achieves the comparable performance with other methods. Specially, on the data sets with high imbalance ratio (e.g., the *balance* data set), the proposed approach can still achieve the good performance and outperforms other methods with large margin. In fact, the experiments have been conducted to obtain the standard deviations which are small (generally below 0.015). Therefore, the differences are statistically significant. Due to the page limitation, the standard deviations are not shown in this paper.

The experiment results can show that the proposed approach outperforms the methods which pre-fix the sampling degree or heuristic rules (*e.g.* Under, SMOTE), and is better than the data-level algorithms. By the multi-classifier boosting algorithm, this cost-sensitive learning method can improve each sub-classifier, and consequently improve the final classifier maximally. Considering the strict and objective evaluation including 3 evaluation metrics, 16 data sets and 12 state-of-the-art compared methods, the validity of the proposed approach can be adequately verified and justified.

Conclusion

In this paper, a new approach and algorithm has been proposed by combining the advantages of data-level and algorithm-level to boost the class-imbalance learning. The proposed approach can adaptively over-sample the minority positive examples and under-sample the majority negative examples to form different sub-classifiers, with the best

cost ratio adaptively chosen for classifiers. The experimental results have shown the proposed approach can achieve the superior performance. In the future, the performance of the proposed approach will be further improved, with more effective yet efficient under-sampling and over-sampling methods.

Table 3: *AUC* of 14 methods.

	SVM	Bagg	Ada	Asym	Under	SMOTE	Chan	RF	BRF	Under-RF	Over-RF	Cascade	Easy	My approach
car	0.991	0.995	**0.998**	**0.998**	0.989	0.995	0.996	0.784	0.749	0.786	0.785	0.996	0.994	0.993
ionosphere	0.980	0.962	0.978	0.979	0.973	0.978	0.979	0.981	0.969	0.976	0.981	0.976	0.974	**0.984**
letter	0.999	0.997	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	0.999	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
phoneme	0.910	0.955	**0.965**	**0.965**	0.953	0.964	0.960	**0.965**	0.960	0.952	0.964	0.962	0.958	0.911
satimage	0.936	0.946	0.953	0.953	0.941	0.946	0.955	0.961	0.952	0.953	**0.962**	0.949	0.947	0.947
wdbc	0.995	0.987	0.994	0.994	0.993	0.994	0.993	0.991	0.990	0.991	0.991	0.994	0.993	**0.995**
abalone	0.776	0.824	0.811	0.812	0.830	0.831	0.850	0.827	0.853	0.842	0.823	0.828	0.847	**0.865**
balance	0.618	0.439	0.616	0.619	0.617	0.617	0.652	0.435	0.558	0.593	0.458	0.637	0.633	**0.890**
cmc	0.692	0.705	0.675	0.675	0.671	0.680	0.696	0.669	0.683	0.676	0.660	0.686	0.704	**0.726**
haberman	0.706	0.669	0.641	0.639	0.646	0.647	0.638	0.645	0.677	0.643	0.641	0.653	0.668	**0.706**
housing	0.801	0.825	0.815	0.815	0.805	0.816	0.811	0.828	0.798	0.820	0.826	0.808	0.825	**0.839**
mf-morph	0.917	0.887	0.888	0.888	0.916	0.912	0.912	0.880	0.901	0.91	0.881	0.905	0.918	**0.931**
mf-zernike	0.900	0.855	0.795	0.801	0.881	0.862	0.903	0.840	0.866	0.889	0.854	0.891	0.904	**0.928**
pima	0.828	0.821	0.788	0.788	0.789	0.792	0.786	0.821	0.809	0.818	0.819	0.799	0.809	**0.828**
vehicle	0.852	0.859	0.854	0.853	0.846	0.858	0.856	0.869	0.850	0.855	0.866	0.856	0.859	**0.879**
wpbc	0.728	0.688	0.716	0.721	0.694	0.709	0.706	0.677	0.646	0.661	0.670	0.712	0.707	**0.728**
average	0.851	0.838	0.842	0.843	0.846	0.850	0.855	0.823	0.828	0.835	0.823	0.853	0.858	**0.884**

Table 4: *F-measure* of 14 methods.

	SVM	Bagg	Ada	Asym	Under	SMOTE	Chan	RF	BRF	Under-RF	Over-RF	Cascade	Easy	My approach
car	0.909	0.933	**0.967**	0.966	0.884	0.930	0.916	0.307	0.521	0.513	0.518	0.945	0.917	0.943
ionosphere	0.926	0.883	0.907	0.910	0.900	0.907	0.910	0.906	0.887	0.895	0.904	0.903	0.903	**0.929**
letter	0.961	0.962	0.988	0.987	0.903	0.954	0.905	0.979	0.889	0.895	0.986	0.979	0.909	**0.990**
phoneme	0.726	0.834	0.850	**0.852**	0.819	0.847	0.837	0.850	0.821	0.813	0.851	0.833	0.822	0.730
satimage	0.582	0.641	0.664	0.668	0.546	0.610	0.607	0.666	0.553	0.557	**0.689**	0.647	0.572	0.607
wdbc	0.965	0.938	0.956	0.956	0.952	0.957	0.954	0.954	0.945	0.948	0.955	0.951	0.951	**0.965**
abalone	0.025	0.170	0.210	0.222	0.367	0.379	0.400	0.189	0.382	0.375	0.253	0.378	0.375	**0.432**
balance	0.000	0.000	0.000	0.000	0.175	0.149	0.156	0.000	0.167	0.168	0.000	0.198	0.161	**0.443**
cmc	0.137	0.362	0.388	0.400	0.429	0.421	0.437	0.347	0.441	0.435	0.408	0.437	0.453	**0.473**
haberman	0.204	0.334	0.348	0.360	0.442	0.405	0.380	0.321	0.468	0.445	0.348	0.431	0.463	**0.470**
housing	0.264	0.419	0.475	0.485	0.529	0.532	0.523	0.445	0.515	0.537	0.490	0.516	0.523	**0.558**
mf-morph	0.011	0.263	0.321	0.344	0.579	0.560	0.635	0.261	0.627	0.602	0.349	0.587	0.623	**0.650**
mf-zernike	0.087	0.183	0.188	0.191	0.538	0.538	0.577	0.144	0.500	0.530	0.292	0.538	0.567	**0.603**
pima	0.612	0.644	0.611	0.613	0.644	0.627	0.618	0.641	0.663	0.668	0.656	0.648	0.654	**0.669**
vehicle	0.477	0.526	0.545	0.561	0.623	0.615	0.608	0.544	0.633	0.633	0.564	0.618	0.637	**0.669**
wpbc	0.301	0.410	0.432	0.444	0.449	**0.459**	0.448	0.393	0.401	0.419	0.397	0.450	0.438	0.396
average	0.449	0.531	0.553	0.559	0.611	0.618	0.619	0.496	0.588	0.589	0.541	0.628	0.623	**0.658**

Table 5: *G-mean* of 14 methods.

	SVM	Bagg	Ada	Asym	Under	SMOTE	Chan	RF	BRF	Under-RF	Over-RF	Cascade	Easy	My approach
car	0.944	0.964	0.980	0.981	0.956	0.969	0.970	0.452	0.693	0.687	0.690	0.980	0.973	**0.982**
ionosphere	0.941	0.906	0.820	0.922	0.918	0.922	0.923	0.918	0.911	0.916	0.918	0.920	0.921	**0.941**
letter	0.972	0.972	0.989	0.988	0.994	0.995	0.992	0.980	0.989	0.993	0.987	**0.996**	0.994	0.988
phoneme	0.796	0.880	0.8901	0.892	0.889	**0.899**	0.897	0.892	0.893	0.887	0.897	0.894	0.892	0.826
satimage	0.703	0.729	0.754	0.761	0.871	0.862	0.881	0.744	0.881	0.883	0.782	0.875	0.887	**0.890**
wdbc	0.972	0.950	0.963	0.963	0.963	0.964	0.962	0.962	0.957	0.960	0.963	0.962	0.963	**0.972**
abalone	0.076	0.337	0.396	0.412	0.765	0.742	0.778	0.363	0.790	0.778	0.457	0.752	0.780	**0.792**
balance	0.000	0.000	0.001	0.002	0.560	0.465	0.465	0.000	0.548	0.548	0.000	0.610	0.580	**0.807**
cmc	0.268	0.509	0.561	0.577	0.623	0.605	0.622	0.516	0.634	0.627	0.587	0.631	0.647	**0.666**
haberman	0.307	0.476	0.502	0.515	0.592	0.562	0.536	0.476	**0.618**	0.593	0.504	0.585	0.611	0.587
housing	0.382	0.553	0.615	0.627	0.725	0.710	0.698	0.580	0.718	0.735	0.638	0.710	0.730	**0.738**
mf-morph	0.018	0.483	0.560	0.594	0.873	0.841	0.920	0.479	0.918	0.888	0.597	0.863	0.914	**0.926**
mf-zernike	0.185	0.378	0.386	0.392	0.848	0.813	0.854	0.326	0.831	0.844	0.519	0.817	0.870	**0.874**
pima	0.690	0.720	0.694	0.696	0.719	0.708	0.700	0.717	0.735	**0.740**	0.731	0.728	0.732	0.730
vehicle	0.588	0.642	0.664	0.679	0.768	0.743	0.738	0.659	0.780	0.779	0.689	0.757	0.780	**0.805**
wpbc	0.378	0.510	0.537	0.549	0.617	0.610	0.585	0.477	0.567	0.588	0.494	**0.630**	0.628	0.598
average	0.513	0.625	0.644	0.659	0.792	0.775	0.782	0.596	0.778	0.777	0.653	0.794	0.806	**0.820**

Acknowledgments

This work was supported by National Hi-Tech Research and Development Program (863 Program) of China under Grants 2014AA015102 and 2012AA012503, National Natural Science Foundation of China under Grant 61371128, and Ph.D. Programs Foundation of Ministry of Education of China under Grant 20120001110097.

References

Batista, G. E.; Prati, R. C.; and Monard, M. C. 2004. A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data. In *Proceedings of ACM SIGKDD Explorations Newsletter* , vol.6, no.1, pp. 20-29.

Breiman, L. 1996. Bagging predictors. *Machine Learning*, vol. 24, no. 2, pp. 123–140.

Breiman, L. 2001. Random forest. *Machine Learning*, vol. 45, no. 1, pp. 5–32.

Cao, Peng; Zhao, Dazhe; and Zaiane, O. 2013. An optimized cost-sensitive SVM for imbalanced data learning. In *proceedings of Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining (PAKDD)*, pp. 280-292.

Chan, P. K., and Stolfo, S. J. 1998. Toward scalable learning with non-uniform class and cost distributions: A case study in credit card fraud detection. In *Proceedings of 4th ACM SIGKDD*.

Chawla, N. V.; Hall, L. O.; Bowyer, K. W. and Kegelmeyer, W. P. SMOTE: Synthetic minority oversampling technique. *Jthe proposednal of Artificial Intelligence Research (JAIR)*, pp. 321–357, 2002.

Chen, C.; Liaw, A.; and Breiman, L. 2004. Using random forest to learn imbalanced data. Dept. Statistics, Univ, California, Berkeley, CA, Tech. Rep. 666.

Chen, Ken; Lu, Bao-Liang; and Kwok, James T. 2006. Efficient Classification of Multi-label and Imbalanced Data Using Min-Max Modular Classifiers. In *Proceedings of International Joint Conference on Neural Networks (IJCNN)*, pp. 1770-1775.

Ertekin, S.; Huang, J.; Bottou, L.; and Giles, C. L. 2007. Learning on the border: active learning in imbalanced data classification. In *Proceedings of Conference on Information and Knowledge Management (CIKM)*, pp. 127-136.

Estabrooks, A.; Jo T.; and Japkowicz N. 2004. A multiple resampling method for learning from imbalanced data sets. *Computational Intelligence*, vol. 20, no. 1, pp. 19-36.

Folleco, Andres; Khoshgoftaar, Taghi M.; and Napolitano, Amri. 2008. Comparison of Fthe proposed Performance Metrics for Evaluating Sampling Techniques for Low Quality Class-Imbalanced Data. In *Proceedings of International Conference on Machine Learning and Applications (ICMLA)*, pp. 153-158.

Frank, A., and Asuncion, A. 1998. UCI Machine Learning Repository. CA: University of California, Irvine, School of Information and Computer Science, http://archive.ics.uci.edu/ml.

Guo, Hongyu, and Viktor, Herna L. 2004. Learning from Imbalanced Data Sets with Boosting and Data Generation The DataBoost-IM Approach. In *Proceedings of ACM SIGKDD Explorations*, vol. 6, no.1, pp. 30-39.

Han, H.; Wang, W.; and Mao, B. 2005. Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning. In Proceedings *of International Conference on Intelligent Computing (ICIC)*, pp. 878-887.

He, Haibo; Bai, Ynag; Garcia, Edwardo A.; and Li, Shutao. 2008. ADASYN Adaptive Synthetic Sampling Approach for Imbalanced Learning. In *Proceedings of International Joint Conference on Neural Networks (IJCNN)*, pp. 1322-1328.

Liu, Xu-Ying; Wu, Jianxin; and Zhou, Zhi-Hua. 2006. Exploratory Under-Sampling for Class-Imbalance Learning. In *Proceedings of International Conference on Data Mining (ICDM)*.

Liu, Xu-Ying; Wu, Jianxin; and Zhou, Zhi-Hua. 2009. Exploratory Undersampling for Class-Imbalance Learning. *IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics*, vol. 39, no. 2, pp. 539-550.

Molinara, M.; Ricamato, M.T.; and Tortorella, F. 2007. Facing imbalanced classes through aggregation of classifiers. In *proceedings of International Conference on Image Analysis and Processing (ICIAP)*, pp. 43-48.

Nguyen, Thai-Nghe; Zeno, Gantner; and Lars, Schmidt-Thieme. 2010. Cost-Sensitive Learning Methods for Imbalanced Data. In *Proceedings of International Joint Conference on Neural Networks (IJCNN)*.

Schapire, R. E. 1999. A brief introduction to boosting. In *Proceedings of 16th International Joint Conferences on Artificial Intelligence (IJCAI)*, pp. 1401–1406.

Sun, Yanmin; Kamel, Mohamed S.; and Wang, Yang. 2006. Boosting for Learning Multiple Classes with Imbalanced Class Distribution. In *Proceedings of International Conference on Data Mining (ICDM)*.

Tang Yuchun; Zhang Yan-Qing; Chawla N.V.; and Krasser S. 2009. *IEEE Transaction on Systems, Man, and Cybernetics (TSMC)*, vol. 39, no. 1, pp. 1083-4419.

Tao, Dacheng ;Tang, Xiaoou; Li, Xuelong; and Wu, Xindong. 2006. Asymmetric Bagging and Random Subspace for Support Vector Machines-Based Relevance Feedback in Image Retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 7, pp: 1088-1099.

Viola, P., and Jones, M. 2002. Fast and robust classification using asymmetric AdaBoost and a detector cascade. In *Proceedings of Neural Information Processing System (NIPS)*, pp. 1311–1318.

Wu, G., and Chang, E. Y. 2003. Adaptive Feature-Space Conformal Transformation for Imbalanced-Data Learning. In *Proceedings of International Conference on Machine Learning (ICML)*, pp. 816-823.

Wu, G., and Chang, E. Y. 2005. KBA: Kernel Boundary Alignment Considering Imbalanced Data Distribution. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, vol. 17, no. 6, pp. 786–795.

Yuan, J.; Li, J.; and Zhang, B. 2006. Learning concepts from large scale imbalanced data sets using support cluster machines. In *Proceedings of ACM Multimedia Conference (MM)*, pp. 441-450.