AN IMBALANCED DATA CLASSIFICATION METHOD BASED ON AUTOMATIC CLUSTERING UNDER-SAMPLING

Xiaoheng Deng, Weijian Zhong, Ju Ren, Detian Zeng
School of Information Science & Engineering
Central South University, Changsha, China, 410083
{dxh,Weijian, ren_ju, zengdetian}@csu.edu.cn

Honggang Zhang
Engineering Department
University of Massachusetts Boston
honggang.zhang@umb.edu

Abstract—Classification of imbalanced datasets has become one of the most challenging problems in big data mining. Because the number of positive samples is far less than the negative samples, low accuracy and poor generalization performance and some other defects always go with learning process of traditional algorithms. Ensemble construction algorithm is an important method to handle this problem. Especially, the ensemble construction algorithm based on random under-sampling or clustering can effectively improve the performance of classification. However, the former causes information loss easily and the latter increases complexity. In this paper, we propose ACUS, an improved ensemble algorithm based on automatic clustering and under-sampling. ACUS conducts clustering first according to the weight of samples, and then it constructs balanced-distributed dataset which consists of a certain percentage of the majority class and all of the minority class from each cluster. With Adaboost algorithm construction, these data-sets are used to get an ensemble classifier. Experimental results demonstrate the advantages of our proposed algorithm in terms of accuracy, simplicity and high stability.

Keywords—Classification; Ensemble; Imbalanced datasets; Class distribution; Boosting

I. INTRODUCTION

In the era of big data, the greater number of noisy data and the more complicated data-distribution bring new challenges to data analytics. Imbalanced data is one of the these challenges, which exists widely in various fields, including medical diagnosis [1], fraud detection [2], detection of oil leakage from the satellite radar image [3], etc.. For these data, the number of negative samples is generally greater than that of positive samples, thus applying standard classifier learning algorithms and evaluation criterion may cause positive samples to be ignored or treated as noise.

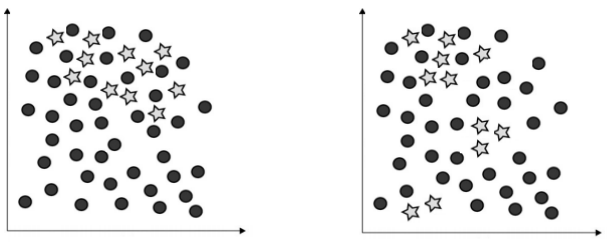


Fig. 1. Example of difficulties in imbalanced data-sets. (a) Class overlapping. (b) Small disjuncts.

The first difficulty of analyzing these data is the imbalanced number of classes. For instance, in a ten-digit recognition problem, the number of samples in each class could be one out of ten of the total number of samples, which leads to a 1:9 imbalance pattern classification problem for each class [4].

Besides, as shown in Fig.1, overlapping and small disjuncts [5] are the other two difficulties of handling im-balanced data. The former makes the normal data submerged easily and the latter complicates the data-distribution. Both of them will seriously degrade the performance of a standard classifier.

In order to deal with the problems of imbalanced data, many solutions have been put forward. However, most of those solutions in general have various problems and should be improved significantly. Therefore, in this paper, we propose an improved ensemble algorithm based on automatic clustering and under-sampling (ACUS), and we show that it deals with the problems of imbalanced data quite well.

This paper is organized as follows. Section II provides a brief review on related work. Section III presents the new method. Experiments based on artificial imbalance datasets and KEEL datasets are shown in Section IV. We conclude this paper in Section 0.

II. RELATED WORK

To address the problems of imbalanced data, researchers have conducted a lot of research and introduced many solutions [6][7][8][9][10], which can be roughly divided into the following classes: data layer based method [6], algorithm layer based method [8][9], the method combining pre-processing on data layer and ensembles [10][11], and the cluster-based under-sampling methods [16][17][18].

The data layer based methods preprocess the data to achieve the effect of balanced data samples as far as they can. SMOTE [6] uses artificial synthesis of new samples in order to balance the sample distribution. However, without taking neighboring samples of the majority class into account, it cannot control the number of synthetic samples precisely, which may cause the overlapping of the samples.

Algorithm layer based methods improve algorithms by considering the distribution of data. The cost sensitive methods focus on the error cost of different situations in the classification process. As the most popular approaches to

constructing ensembles by data variation, AdaBoost [8] and Bagging [9] integrate the base classifiers in a specific way, which solves the problem of imbalanced datasets to a certain extent. Nevertheless, ensembles are not able to deal with imbalanced problems on their own, since they are inherently designed to maximize accuracy of total samples.

In addition, algorithm layer based methods are combined with preprocessing algorithms and ensembles framework to address the problem of uneven class distributions. Among these methods, SMOTEBoost [10] combines SMOTE and Adaboost to introduce the synthetic samples in each iteration to make classifiers pay attention to the classes equally. RUSBoost [11] combines under-sampling and Adaboost, producing balanced dataset that is composed of a portion of extracted samples from majority class and all the samples from minority class. However, the process of sample extraction is random so that it will result in the information loss of the majority class. Even though the boosting method makes up for the information loss, the extent of the loss is random and cannot be controlled.

Cluster-based under-sampling was introduced in [16][17][18] to reduce information loss. These methods find the most suitable training samples by clustering as far as possible. EHCU [16] uses hierarchical clustering method, and the sampling is restricted to some negative samples which are close to the positive samples. Although clustering methods reduce information loss and improve the performance of classification, with the completion of the clustering algorithm, the subset will be fixed, which lacks flexibility. On the other hand, clustering algorithms always have to calculate the distance of all samples, which bring a very large computational overhead.

To address the limitations mentioned above, we introduce in this paper a new method that combines the way that Adaboost algorithm modify the weight of samples and the idea of data clustering. Our method splits the negative samples into some small sub-clusters on the basis of samples weight. These sub-clusters have small statistical correlation with each other so that we can easily sample them to compose a representative training set with all the positive samples. Then, the random under-sampling is optimized to improve the classification accuracy with only a small increase of computation overhead.

III. PROPOSED METHOD

It is crucial to select an effective sample to construct our training sample set. Although some scholars put forward the method of clustering based under-sampling [16][17][18], that method lacks flexibility, with a lot of iterative distance calculation. To solve that problem, we propose an automatic clustering algorithm based on sample weight, abbreviated as ACUS.

Our ideas is summarized as follows: 1) Selecting samples from different clusters is helpful for focusing on important samples. 2) Use variance to determine if a cluster could be divided. 3) Determine the importance of a cluster by its weight so that we can find the important samples.

A. Weight Variance Based Cluster Segmentation

A good sampling should present the information of the negative samples as much as possible. Therefore, in order to improve under-sampling, the set of negative samples is divided into comparatively ordered clusters which have small statistical correlation with each other. Then a new training set is constructed, which is made up of a certain number of extracted samples from the clusters and all the positive samples.

In order to characterize the separability of the samples better, we set the weight for each sample like what AdaBoost algorithm [8] does. These weights can be adjusted after every classifier is learned. In the Adaboost algorithm, suppose $h(x_i)$ represents a weak classifier, w represents the sample weight, and y represents the class of samples, then the method is shown in Eq. (1).

$$w_{h(x_i)\neq y_i} = w_{h(x_i)\neq y_i} \times \frac{1-\varepsilon}{\varepsilon}, \qquad \varepsilon = \sum_{h(x_i)\neq y_i} w_{h(x_i)\neq y_i} \tag{1}$$

By modifying the weights in each round of iteration, the samples which are difficultly classified correctly often have a relatively high sample weight. On the contrary, the weight of the samples which can be classifier correctly easily is lower. Make *wvar* specify the weight variance of the cluster. When there is a cluster called C, the number of samples in C is NC and each weight of sample is w_k, the variance of weight is as Eq.(2).

$$wvar(C) = \sum_{i=0}^{NC}\left(w_k - \frac{\sum_{k=0}^{I} w_k}{NC}\right)^2 \Big/ NC \tag{2}$$

Variance is used in Regression Trees algorithm [12] that if an attribute has large variance, it is more likely the larger range of values will capture the variability of the classes better.

By using the weight variance in this paper, the sample order can be evaluated. If the weight of samples in a cluster is approximate, the weight variance of the cluster is small and the samples in the cluster are orderly. Therefore, we aim to divide a cluster with high *wvar* into two more orderly clusters to decrease the variance. Divide C into two sub-clusters, namely C_1, C_2 and specify the weight variance of sub-clusters according to Eq.(3)

$$wsvar(C_1, C_2) = \frac{size(C_1)}{size(C)} \times wvar(C_1) + \frac{size(C_2)}{size(C)} \times wvar(C_2) \tag{3}$$

where *size* represents the number of samples in the collection. If the samples' weights in the two sub-clusters C_1, C_2 are more orderly than the original cluster C, the *wsvar* is smaller than *wvar*. When the number of samples in the cluster C is NC, the number of possible sub-clusters is NC-*1*. The details of selecting optimal sub-clusters is given in Algorithm 1.

Algorithm 1 Division of a cluster

Input: *C*: a cluster of negative samples; *NC*: the number of samples in the cluster *C*; w_k: the weights of sample in *C*, *k=1,2…NC*

Output: two sub-clusters C_1, C_2

Set coefficient of partition α

1. Calculate the variance of weight $wvar_s = wvar(C)$
2. sort sample according to the weights of the samples in *C*.
3. go through the samples in *C* and calculate the *wsvar* in each loop to get the minimum $wsvar$.

$$wsvar_{min} = min_{0<k<NC}\{wsvar(C_1', C_2')\}$$

4. If $wsvar_{min} < \alpha \cdot wvar_s$: divide the original cluster into two sub-clusters P_1 and P_2, all weights of the samples are normalized.

B. Under-sampling From The Clusters

In order to get a strong representative sample from the clusters, this paper considers two factors, which are the quantity of sampling and the sampling method.

The number and the value of samples in the cluster are different. For the sake of the comprehensive consideration, this paper uses the sum of cluster sample weight as the reference index to determine the sample number of each cluster. A sample weight of a cluster C_j is w_k and the number of negative samples is *N*, then the sampling number of C_j is SNC_j shown in eq.(4).

$$W = \sum_{i=1}^{N} w_i$$
$$SNC_j = \sum_{w_k \epsilon C_j} w_k / W \quad (4)$$

Our method ensures that a small number of noise samples with higher weight values will not affect the overall sampling. Thus, the algorithm focuses on the most important samples and eliminates the noise points effectively.

In order to pay more attention to the important samples, our sampling method takes the top SNC_j weight of sampling in each cluster as training samples. This method is better than the TOP-K algorithm, which does not introduce significant computational complexity.

C. Combining Boosting And Automatic Clustering

Due to the diversity they can provide, random techniques are powerful when constructing ensembles. Combined with accurate base classifiers, random techniques can produce high performance ensembles.

However, we think that an uncontrolled randomness should be closely managed when dealing with highly imbalanced datasets, in order to get an improved performance. Even though random under-sampling is often an appropriate technique for assessing good results, it might discard potentially useful instances of the majority class, which might be important for the learning process. Besides, as the IR of the dataset increases, so does the probability of ignoring useful majority class examples. For this reason, we focus on highly imbalanced datasets and solve this problem by using ACUS.

The inclusion of ACUS within Boosting algorithm is simple and easy to implement, yet effective. We borrow the idea of RUSBoost [11] and other Boosting-based algorithms [13], and introduce the under-sampling process inside the loop of AdaBoost. In this case, ACUS is used instead of random under-sampling or SMOTE, and then, only the weights of the instances in the under-sampled dataset are used in the division of clusters. The whole procedure of ACUS is outlined as Algorithm 2.

Algorithm 2

Input: Training set $S=\{x_i,y_i\}$, *i=1, 2…N;* S^+: the Minority class set; S^-:the majority class set;S^+,$S^- \epsilon S$; C_j: the clusters of S^-; *J*: the number of clusters; *I*: WeakLearn

Setting: *MN*: Maximum number of clusters; *step*: step for cluster partitioning

1: Initialize samples weight: $w_i^1 = 1/N$ for i=1, ..., N

2: Let $C_1=S^-$

3: Do for *t=1,2 …T*

a. Determine the sampling number of each cluster SNC_j using eq. (4)

b. Take the top *SNCj* weight of samples in each cluster as training samples and combine with S^+ to get the temporary training set S' and its weight w'

c. Call WeakLearn, providing it with w'; get back a hypothesis $h(t) \leftarrow I(S')$

d. Calculate the error of

$$h(t): \varepsilon_t = \sum_{jy_i \neq y_{ji}} \left(1 - h_t(x_i, y_i) + h_t(x_i, y)\right)$$

e. Set $\beta = \varepsilon_t/(1-\varepsilon_t)$

Set the new weights vector to be

$$w_i^{t+1} = w_i^t \, \beta_t^{1-[\![h_t(x_i) \neq y_i]\!]}$$

f. if *t % step==0 and NS<MN*

for *j=1,..,J* ,divide C_j , by using *Algorithm 1*

Output: the hypothesis:

$$H(x) = argmax \sum_{t=1}^{T} h_t(x,y) log\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$$

The algorithm of ACUS is based on the framework of Adaboost algorithm, in which the weights of the samples are modified and the classifier is trained. After a certain number of iterations, the clusters of negative samples are finally divided into some sub-clusters until the number of clusters is not less than *MN*. Before training a new classifier, the samples are extracted according to the weights of the samples in the clusters. Compared with the traditional method, this method can better detect the representative samples without calculating the complex distance.

D. Time Complexity of ACUS

The ACUS consists of three major steps: 1) clustering of samples in the majority class, 2) sampling from clusters, and 3) training the ensemble classifier. Suppose that the number of positive samples is N_p, and the number of negative samples is N_n. The time complexities of those three steps are $O(N_n \log(N_n) t)$, $O(N_p \log N_n)$, and $O(tB)$, respectively, where t, d, and B denote the number of steps used by clustering, the number of input features, and the complexity of the base classifier respectively. The time complexity of K-Means and hierarchical clustering are $O(Ntd)$ and $O(N^2 dlogN)$, respectively, where $N=N_n+N_p$. Thus the time complexity of ACUS is not worse than K-Means but better than hierarchical clustering in clustering procedures. In addition, ACUS yields significantly better results.

IV. Experiment and Analysis

In this section, the algorithm presented in this paper and some state-of-the-art algorithms for imbalanced data sets, such as RUSboost, SMOTEBOOST, would be compared in order to test the effectiveness of the proposed algorithm. The specific situation of the algorithms on experimental comparison is shown in Table 1.

In order to demonstrate the effect of clustering, sampling and integration, our experiment is divided into two parts. Firstly, we use three artificial imbalance datasets to simulate different distribution of unbalanced data samples, on which cluster and sampling experiments are carried out. Secondly, AUC [14] and KAPPA [15] would be used to verify the accuracy and stability of the algorithm based on the 22 data sets of KEEL, which are publicly available on the corresponding web-page.[1]

We use CART algorithm [12] as the base classification in the experiments. The α for algorithm 1 is set to be 0.27 while the *step* is 5 and the *MN* used in algorithm 2 is set to be the ratio of the imbalance.

A. Experiment Based On Artificial Imbalance Datasets

1) Samples clustered by different clustering schemes

In order to verify the effectiveness of the proposed clustering method, we compare the proposed method with other algorithms. This paper uses the clustering algorithm including K-means clustering algorithm [17][18] and hierarchical clustering algorithm [16].

[1] http://www.keel.es/dataset.php.

Table 1 State-of-the-art ensembles considered

Abbr.	Short description
RUS	AdaBoost.M2 with random under-sampling
SMO	AdaBoost.M2 with SMOTE
CEU	AdaBoost with under sampling based on K-Means Clustering
EHCU	Bagging with under sampling based on Hierarchical Clustering
ACUS	AdaBoost with under sampling based on automatic clustering

The three 2-D artificial datasets, shown in Fig. 2, represent three different situations of imbalance datasets. Each dataset has 1000 samples with a ratio of 1:9 between minority and majority classes. Fig. 2(a) shows the first imbalanced dataset which has the obvious boundary between minority class and majority class. The second imbalanced dataset is an overlapping situation, as shown in Fig. 2(b). Small disjuncts are presented in the third imbalanced dataset, shown in Fig. 2(c).

Fig.3-5 show the clustering results of the three algorithms on the three artificial datasets, in which the sample points with same color are in the same cluster. 'x' means that the samples are in the majority class. 'o' means that the samples are in the minority class.

As shown in Fig. 3(a), when the overlapping of the samples is not significant, ACUS can distinguish those negative samples that are closer to the positive samples. And it is good at screening out those unimportant negative samples as noise samples by dividing them into separate several clusters. When the sample overlapping degree is high, as shown in Fig.3(b), the proposed algorithm can identify those overlapped samples and differentiate them from other samples. Fig. 3(c) shows that when small disjuncts occurs, ACUS is concerned about those negative samples which are between the sub-clusters and important for classification.

Fig. 4 shows that K-means clustering is more likely to cluster the samples by spatial distribution. It achieves good result when the sample overlapping degree is not high. If the high overlapping or small disjuncts occurs, the solution of K-Means is not targeted.

The results of hierarchical clustering are shown in Fig. 5, in which the light gray 'x' indicates that the unimportant negative samples detected by the algorithm. The results show that hierarchical clustering eliminates unimportant negative samples well, though it pays too much attention to noise samples which can easily lead to over fitting when the ratio of overlapping or the signal to noise is high.

2) Samples selected by different clustering schemes

This part discusses the experimental results of sampling by those three algorithms. Fig. 7 shows the sampling results of the method used by CEU, which samples after K-Means clustering. The results of distance based sampling after hierarchical clustering [16] is shown in Fig. 8.

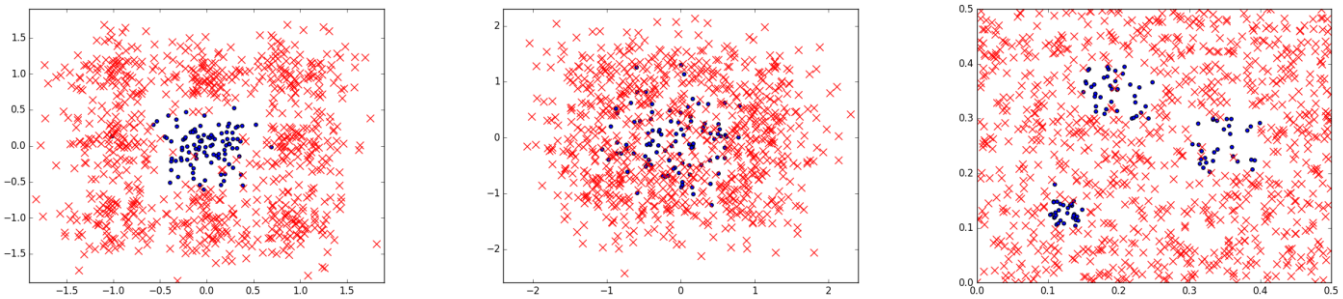
The experimental results show that K-Means and sample-weight based under-sampling algorithms cannot get good result or control the overlapped noise well. On the contrary, hierarchical clustering based method can achieve good performance when the degree of overlap is not high. But it

focuses too much on the overlapping part of the sample easily when the samples are overlapped.

Compared with them, our proposed method has good performance on those three datasets, as shown in Fig. 6. When sample overlapping degree is high, it can ignored most of the overlapped negative samples. And when small disjuncts occurs, it has good performance in terms of identifying class boundaries.

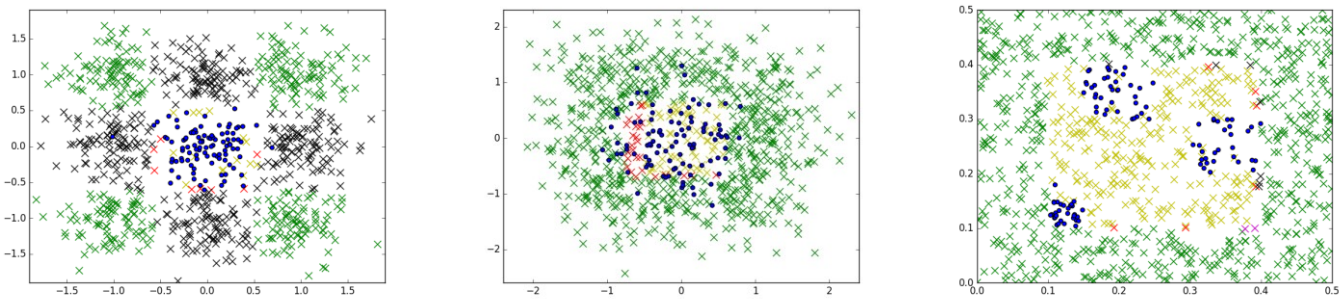
3) Classification results on artificial datasets

This section shows the performance of CEU, EHCU and ACUS on three different types of imbalanced data in Fig.1 by using the AUC as the evaluation index. The results of the experiment are shown in Table 2, which indicate EHCU cannot deal with the overlapping problem well and CEU does not bring too much improvement. On the contrary, ACUS has the better performance on those three imbalanced datasets. This shows that the ACUS is more efficient and effective in selecting useful samples from both the positive and negative samples.



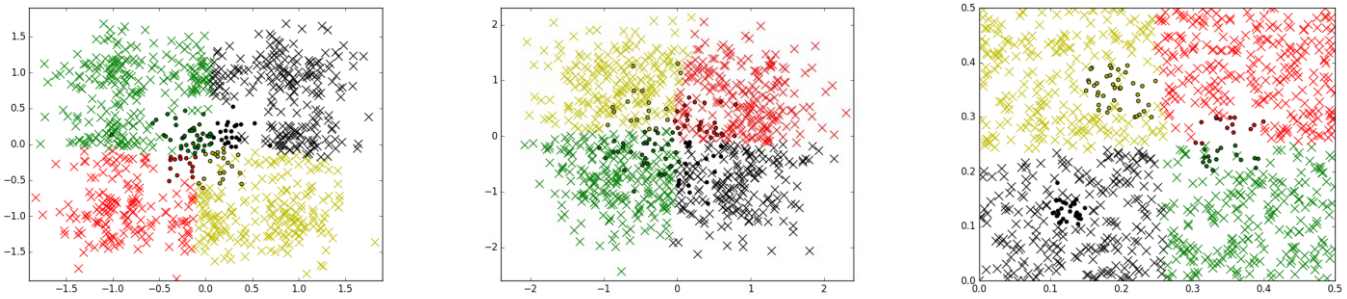
(a) (b) (c)

Fig. 2. Artificial imbalance datasets



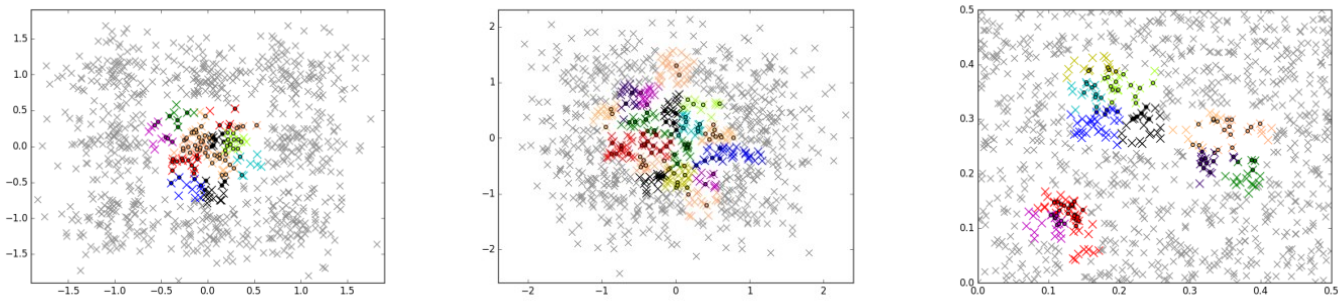
(a) (b) (c)

Fig. 3. Weight variance based clustering by ACUS



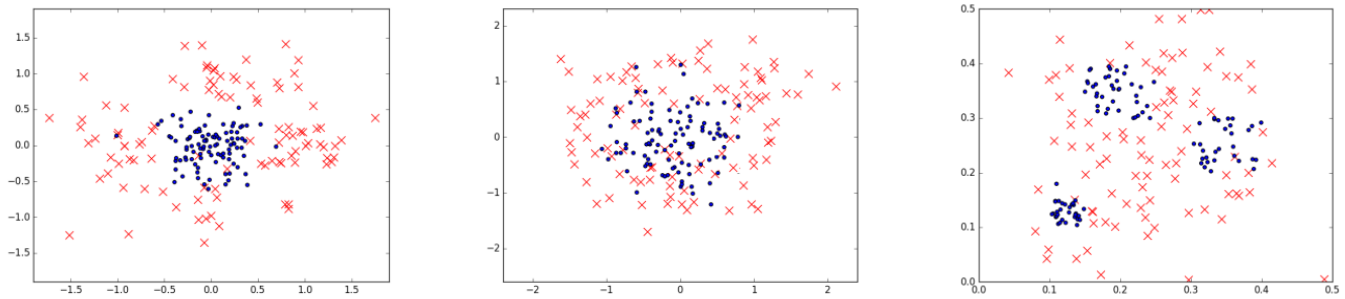
(a) (b) (c)

Fig. 4. K-Means clustering by CEU



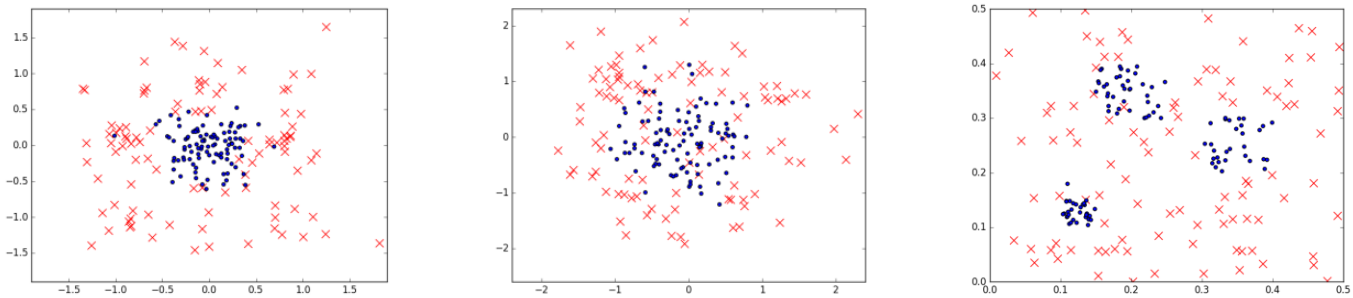
(a) (b) (c)

Fig. 5. Hierarchical clustering by EHCU



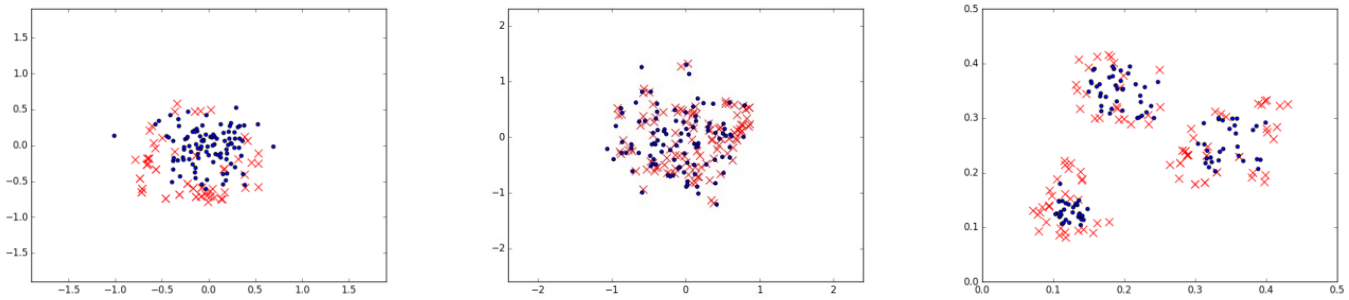
(a) (b) (c)

Fig. 6. Top-K under-sampling by ACUS



(a) (b) (c)

Fig. 7. Sample weights based under-sampling by CEU



(a) (b) (c)

Fig. 8. Nearest distance under-sampling by CHCU

Table 2 result of AUC metric on artificial data-sets

datasets	CEU	EHCU	ACUS
1 normal	0.9946	0.9832	0.9876
2 overlapping	0.7874	0.6454	0.9037
3small disjuncts	0.8064	0.9012	0.9300

Table 3 Summary description of the imbalanced data-sets.

No.	Data-sets	#Ex.	#Atts.	IR
1	yeast-2_vs_4	514	8	9.089
2	yeast-0-5-6-7-9_vs_4	528	8	9.359
3	vowel0	988	10	10.109
4	glass-0-1-6_vs_2	192	9	10.299
5	glass2	214	9	10.399
6	ecoli4	336	7	13.84
7	shuttle-c0-vs-c4	1829	9	13.879
8	yeast-1_vs_7	459	7	13.879
9	glass4	214	9	15.479
10	page-blocks-1-3_vs_4	472	9	15.857
11	abalone9-18	731	8	16.68
12	glass-0-1-6_vs_5	184	9	19.449
13	shuttle-c2-vs-c4	129	9	20.59
14	yeast-1-4-5-8_vs_7	693	8	22.109
15	glass5	214	9	22.819
16	yeast-2_vs_8	482	8	23.109
17	yeast4	1484	8	28.419
18	yeast-1-2-8-9_vs_7	947	8	30.569
19	yeast5	1484	8	32.789
20	ecoli-0-1-3-7_vs_2-6	281	7	39.159
21	yeast6	1484	8	39.159
22	abalone19	4174	8	128

B. Experiment Based On KEEL Datasets

In this section, ACUS is compared with the state-of-the-art ensemble methods based on 22 KEEL Datasets, which are shown in Table 3. In order to obtain these two-class imbalanced problems, original multi-class datasets were modified in such a way that the union of one or more classes was labeled as the positive class, and the same was done to obtain the negative class. Table 3 summarizes the properties of the datasets: the number of examples (#Ex.), number of attributes (#Atts.), and Imbalance Ratio (IR). This table is sorted according to IR.

1) Experimental results of AUC metric

We have obtained AUC metric estimates using a 5-fold stratified cross-validation. This process was carried out three times with different seeds and we take the average as the final result, which are shown in Table 4.

The result shows that our proposed method yields the best results in most datasets in terms of AUC. Compared with RUSBoost, its average result of AUC is improved by 4% approximately. These results show that the ACUS performs well across different feature types, imbalance ratios, numbers of feature, and numbers of sample. Thus, our proposed method has better classification performance compared with some state-of-the-art methods.

2) Experimental results of KAPPA metric

Because of the instability of undersampling, the classification performance of RUSBoost is not stable. In order to measure the stability, this paper considers kappa coefficient [15] to measure the stability of the classifier. In this experiment, we divide each dataset into a training set and a test set. Each of the algorithms is run 5 times based on the training sets, and then the kappa coefficient of each two classification results on test sets is measured. Thus, there are 10 experimental results for each algorithm.

Table 5 shows the average value of each algorithm with their 10 experimental results, which indicates that ACUS has better overall stability than others. Its average stability increase is 12% compared with RUSBoost and 2% compared with CEU.

Table 4 result of AUC metric

Data-sets	IR	RUS	SMO	CEU	EHCU	ACUS
yeast-2_vs_4	9.089	**0.9706**	0.9454	0.9591	0.9565	0.9691
yeast-0-5-6-7-9_vs_4	9.359	0.8497	0.8341	0.8466	0.7737	**0.8766**
vowel0	10.109	0.9774	0.9888	0.9795	**0.9919**	0.9815
glass-0-1-6_vs_2	10.299	0.6967	0.7657	0.8043	0.6348	**0.8129**
glass2	10.399	0.7270	0.7612	**0.8785**	0.5959	0.8697
ecoli4	13.84	0.9389	0.9149	0.9643	**0.9995**	0.9643
yeast-1_vs_7	13.879	0.7369	0.7321	0.7943	0.7694	**0.8481**
shuttle-c0-vs-c4	13.879	1.0000	1.0000	1.0000	1.0000	**1.0000**
glass4	15.479	**0.9670**	0.9207	0.9504	0.6515	0.9479
page-blocks-1-3_vs_4	15.857	0.9990	1.0000	1.0000	0.9710	**1.0000**
abalone9-18	16.68	0.7871	0.7599	0.8038	0.6973	**0.8648**
glass-0-1-6_vs_5	19.449	0.9914	0.9871	0.9933	0.9576	**0.9971**
shuttle-c2-vs-c4	20.59	1.0000	1.0000	1.0000	1.0000	**1.0000**
yeast-1-4-5-8_vs_7	22.109	0.6384	0.6165	0.6484	0.5474	**0.6700**
glass5	22.819	0.9915	0.9788	1.0000	0.9593	**1.0000**
yeast-2_vs_8	23.109	0.7436	0.7930	0.7395	**0.8687**	0.7821
yeast4	28.419	0.8909	0.8804	0.8835	0.7277	**0.9216**
yeast-1-2-8-9_vs_7	30.569	0.6997	0.6712	0.7033	0.7052	**0.7955**
yeast5	32.789	0.9802	0.9793	0.9883	**0.9961**	0.9926
ecoli-0-1-3-7_vs_2-6	39.159	0.7975	0.8795	0.8024	0.8891	**0.9444**
yeast6	39.159	0.9009	0.8910	0.9233	0.8440	**0.9524**
abalone19	128	0.6853	0.6755	0.7589	0.6863	**0.8066**
Average		0.8623	0.8625	0.8828	0.8283	**0.9090**

Table 5 result of KAPPA metric

Data-sets	IR	RUS	CEU	ACUS
yeast-2_vs_4	9.089	**0.8964**	0.8886	0.8559
yeast-0-5-6-7-9_vs_4	9.359	0.7004	0.8703	**0.978**4
vowel0	10.109	**0.9964**	0.9545	0.9512
glass-0-1-6_vs_2	10.299	**0.8619**	0.7619	0.8002
glass2	10.399	0.7918	0.7845	**0.8603**
ecoli4	13.84	0.9524	0.9030	**1.0000**
yeast-1_vs_7	13.879	0.2210	0.5523	**0.6277**
shuttle-c0-vs-c4	13.879	0.9832	**0.9989**	0.9901
glass4	15.479	0.9275	0.9530	**0.9569**
page-blocks-1-3_vs_4	15.857	**1.0000**	**1.0000**	**1.0000**
abalone9-18	16.68	0.7593	0.7431	**0.8992**
glass-0-1-6_vs_5	19.449	0.9145	0.9943	**0.9971**
shuttle-c2-vs-c4	20.59	**1.0000**	**1.0000**	**1.0000**
yeast-1-4-5-8_vs_7	22.109	**0.6847**	0.6754	0.6769
glass5	22.819	**0.9153**	0.8279	0.8768
yeast-2_vs_8	23.109	**0.6951**	0.6429	0.6414
yeast4	28.419	0.4566	0.7139	**0.7231**
yeast-1-2-8-9_vs_7	30.569	0.0624	0.7377	**0.8857**
yeast5	32.789	0.8842	0.8822	**0.9906**
ecoli-0-1-3-7_vs_2-6	39.159	0.7262	**0.9945**	0.9318
yeast6	39.159	0.7279	0.8971	**0.9298**
abalone19	128	-0.0053	0.4932	**0.5038**
Average		0.7342	0.8430	**0.8671**

V. CONCLUSION

In this paper, we proposed ACUS, which use samples' weights to make clustering automatically and efficiently. It preserves the distribution information of the majority class and selects informative samples from the majority class. Based on artificial imbalance datasets, experimental results show that the ACUS is robust to different distributions of unbalanced data samples. And its average result increase by 2% when compared with the second best state-of-the-art methods. Those results show that ACUS can achieve good classification result easily. One future work will be on the re-combination of clusters, which might make more reasonable clustering and take more potentially useful instances of the majority class into consideration. Moreover, when data changes over time, incremental learning is needed. Very few work have been done in dealing with imbalance problems in incremental learning, which is an important issue for real-world application.

VI. ACKNOWLEDGEMENTS

The author gratefully acknowledges support from National Natural Science Foundation of China projects of grant No. 61272149, 61379058, 61379057, 61350011.

REFERENCES

[1] He H., Garcia E. A. Learning from imabalanced data[J]. IEEE Transactions on Knowledgeand Data Engineering. 2009, 21(9):1263-1284

[2] Chan P.K., Stolfo S.J. Toward Scalable Learning with Non-Uni-form Class and Cost Distributions: A Case Study in Credit Card Fraud Detection[C] KDD.1998:164-168

[3] Kubat M., Holte Rc, Matwin S. Machine learning for the detection of oilspillsinsatel literadarima. 1998,30(2/3):195-215

[4] J. C. Fernandez Caballero, F. J. Martínez, C. Hervás, and P. A. Gutiérrez, "Sensitivity versus accuracy in multiclass problems using memetic Pareto evolutionary neural networks," *IEEE Trans. Neural Netw.*, vol. 21,no. 5, pp. 750–770, May 2010.

[5] G. M. Weiss and F. Provost, "Learning when training data are costly: The effect of class distribution on tree induction," *J. Artif. Intell. Res.*, vol. 19,pp. 315–354, 2003.

[6] Lusa L. SMOTE for high-dimensional class-imbalanced data[J]. BMC bioinformatics, 2013, 14(1): 1-16

[7] Elkan C. The foundations of cost-sensitive learning[C]//In Proceedings of International joint conference on artificial intelligence. LAWRENCE ERLBAUM ASSOCIATES LTD, 2001, 17(1): 973-978.

[8] Y. Freund, R.E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, Journal of Computer and System Sciences 55 (1) (1997) 119–139.

[9] L. Breiman, Bagging predictors, Machine Learning 24 (1996) 123–140.

[10] N.V. Chawla, A. Lazarevic, L.O. Hall, K.W. Bowyer, SMOTEBoost: improving prediction of the minority class in boosting, in: Knowledge Discovery in Databases (PKDD'03), 2003, pp. 107–119.

[11] C. Seiffert, T. Khoshgoftaar, J. Van Hulse, A. Napolitano, RUSBoost: a hybrid approach to alleviating class imbalance, IEEE Transactions on Systems, Man,and Cybernetics, Part A: Systems and Humans 40 (1) (2010) 185–197.

[12] Yeh C H. Classification and regression trees (CART)[J]. Chemometrics and Intelligent Laboratory Systems, 1991, 12(1): 95-96.

[13] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, F. Herrera, A review onensembles for the class imbalance problem: bagging-, boosting-, and hybridbased approaches, IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 42 (4) (2012) 463–484.

[14] DItterrich TG.Machine learning research:four current direction [J].Artificial Intelligence Magzine,1997,18(4):97-136

[15] Mc Hugh M L. Interrater reliability: the kappa statistic[J]. Biochemia Medica, 2012, 22(3): 276-282.

[16] Sima Soltani, Javad Sadri, Feature Selection and Ensemble Hierarchical Cluster-based Under-sampling Approach for Extremely Imbalanced Datasets[J] International eConference on Computer and Knowledge Engineering (ICCKE) (2011) 13-14,

[17] Zhang X S, Luo Q. Unbalanced Data Classification Algorithm Based on Clustering Ensemble Under-sampling[J]. Computer Science, 2015, 42(11): 63:66

[18] W. W. Y. Ng, J. Hu, D. S. Yeung, S. Yin and F. Roli, "Diversified Sensitivity-Based Undersampling for Imbalance Classification Problems," in IEEE Transactions on Cybernetics, vol. 45, no. 11, pp. 2402-2412, Nov. 2015.