Information Sciences 477 (2019) 47–54

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Under-sampling class imbalanced datasets by combining clustering analysis and instance selection

Chih-Fong Tsai[a], Wei-Chao Lin[b,c,d,*], Ya-Han Hu[e,f], Guan-Ting Yao[a]

[a] *Department of Information Management, National Central University, Zhongli, Taiwan*
[b] *Department of Information Management, Chang Gung University, Taoyuan, Taiwan*
[c] *Healthy Aging Research Center, Chang Gung University, Taoyuan, Taiwan*
[d] *Department of Thoracic Surgery, Chang Gung Memorial Hospital, Linkou, Taiwan*
[e] *Department of Information Management, National Chung Cheng University, Chiayi, Taiwan*
[f] *Center for Innovative Research on Aging Society, National Chung Cheng University, Chiayi, Taiwan*

ARTICLE INFO

Article history:
Received 2 January 2018
Revised 19 October 2018
Accepted 20 October 2018
Available online 22 October 2018

Keywords:
Data mining
Class imbalance
Clustering
Ensemble classifiers
Instance selection

ABSTRACT

Class-imbalanced datasets, i.e., those with the number of data samples in one class being much larger than that in another class, occur in many real-world problems. Using these datasets, it is very difficult to construct effective classifiers based on the current classification algorithms, especially for distinguishing small or minority classes from the majority class. To solve the class imbalance problem, the under/oversampling techniques have been widely used to reduce and enlarge the numbers of data samples in the majority and minority classes, respectively. Moreover, the combinations of certain sampling approaches with ensemble classifiers have shown reasonably good performance. In this paper, a novel undersampling approach called cluster-based instance selection (CBIS) that combines clustering analysis and instance selection is introduced. The clustering analysis component groups similar data samples of the majority class dataset into 'subclasses', while the instance selection component filters out unrepresentative data samples from each of the 'subclasses'. The experimental results based on the KEEL dataset repository show that the CBIS approach can make bagging and boosting-based MLP ensemble classifiers perform significantly better than six state-of-the-art approaches, regardless of what kinds of clustering (affinity propagation and *k*-means) and instance selection (IB3, DROP3 and GA) algorithms are used.

1. Introduction

In the current era of big data, data mining and analysis are becoming increasingly important for making effective decisions. Among the various data mining techniques, classification analysis is one of techniques most widely used for various business and engineering problems, such as bankruptcy prediction [21], cancer prediction [18], churn prediction [23], face detection [35], fraud detection [31], and software fault prediction [24].

[*] Corresponding author at: Department of Information Management, Chang Gung University, Taoyuan, Taiwan.
E-mail address: viclin@gap.cgu.edu.tw (W.-C. Lin).

https://doi.org/10.1016/j.ins.2018.10.029
0020-0255/© 2018 Elsevier Inc. All rights reserved.

In general, the developed classifiers (or prediction models) usually perform well over evenly distributed data of different classes. However, in practice, the data collected for training the classifiers are usually class imbalanced, i.e., the numbers of data samples in different classes are highly different. In the example of a two-class dataset, the two classes contain 10 and 1000 data samples. In particular, the distribution of data in the feature space is usually skewed in class-imbalanced datasets [8]. Furthermore, datasets with a skewed distribution will usually have some other problematic characteristics, such as data sample overlap, small sample sizes, and small disjuncts [4,13].

The characteristics of class-imbalanced datasets mentioned above differ from the assumption of a relatively balanced distribution of data for most classification algorithms. This difference means that it is very difficult for classifiers to correctly predict the small (or minority) class, and they are likely to misclassify the testing samples into the prevalent (or majority) class [8,28]. However, in many real-world problems, e.g., credit card fraud detection (non-fraud vs. fraud cases), bankruptcy prediction (non-bankrupt vs. bankrupt cases), and various disease detection predictions (non-infected vs. infected cases), the accuracy of detection, prediction or classification of the data in the minority class is critical.

In the literature, three types of approaches have been used to tackle the class imbalance problem. They are the data-level [1,19], algorithm-level [34,36], and cost-sensitive methods [8,22]. Among these, the data-level methods are most widely used for class-imbalanced datasets [13].

The data-level methods aim at reducing the imbalance ratio between the majority and minority classes by either undersampling the data in the majority class [19,33] or oversampling the data in the minority class [1,7]. As the dataset size has been steadily increasing, the undersampling approach should be a better choice than the oversampling approach.

To reduce the number of data samples in the majority class, cluster-based sampling methods were introduced. Such methods can outperform the random sampling approach [20,29,33]. In general, the cluster-based sampling methods are based on grouping a number of clusters from a given majority class dataset, after which a number of representative data samples are selected from each of the clusters. However, there are several limitations of the cluster-based sampling methods that directly affect the reduced majority class dataset and the final classification performance. For instance, it is difficult to decide on the number of clusters needed for the optimal clustering result. In addition, the representative data samples to be selected from each cluster need to be carefully defined. In other words, the original data distribution in the majority class may be changed.

In the related studies of data preprocessing, instance selection is used to filter out unrepresentative data samples (or outliers) from a given training dataset, which can make the classifiers outperform those trained on the original training dataset without performing instance selection [14]. In general, this method can be used to reduce the dataset size of the majority class. However, since the existing instance selection algorithms are designed to distinguish between good and noisy data samples from the multiclass datasets, they cannot handle datasets that only contain one class, i.e., the majority class dataset.

In this study, we present a novel approach called cluster-based instance selection (CBIS) that is derived by combining clustering analysis and instance selection techniques. The characteristics of the clustering analysis and instance selection techniques mean that they complement each other for effective undersampling of the majority class dataset. CBIS is a two-step approach that first uses a clustering technique to group a number of data samples in the majority class, where each data sample belongs to a specific cluster. In particular, each cluster can be regarded as a ‘subclass’ of the majority class. Each data sample is then associated with a new class label, and as a result, a multiclass dataset for the majority class dataset is produced. Next, the instance selection technique is performed over the generated multiclass dataset to reduce the dataset size for the undersampling purpose. Our experimental results obtained with various domain datasets show that the proposed CBIS approach performs better than many state-of-the-art data-level approaches.

The rest of this paper is organized as follows. Section 2 gives an overview of the class imbalance problem and several representative data-level methods. In addition, the clustering analysis and instance selection techniques are also briefly described. Section 3 introduces the proposed CBIS approach. A description of the experimental setup and results is given in Section 4. Finally, Section 5 concludes the paper.

2. Literature review

2.1. Class-imbalanced datasets

In class-imbalanced datasets, the imbalance ratio between the minority and the majority classes can be as drastic as 1:100, 1:1000 or even larger [8]. Although there is no exact answer to the question of what magnitude of imbalance ratio will lead to a deterioration of classification performance, in some applications a ratio of 1:35 can render some methods inadequate for building an effective classifier, while in other cases, a ratio of 1:10 is difficult to deal with [28].

In addition to the skewed distribution, there are three characteristics of imbalanced datasets: class overlap, small sample sizes, and small disjuncts [4,13]. Class overlap means that a number of data samples belonging to different classes overlap in the feature space. For small sample sizes, this means that there will be an insufficient number of data samples in the minority class. Finally, small disjuncts consisting of few examples are extremely error-prone, because the concept of the minority class is formed of subconcepts, i.e., a number of groups of data samples for the same minority classes are distributed separately in the feature space, which increases the complexity of developing an effective classifier.

2.2. Sampling methods

Sampling (or resampling) methods are one type of the state-of-the-art solutions for dealing with class-imbalanced datasets. Sampling methods focus on generating a new dataset from a given class-imbalanced dataset that will contain a more balanced distribution between the majority and minority classes. In most cases, a balanced dataset can provide improved overall classification performance compared to that of an imbalanced dataset [16]. The primary advantage of these sampling methods is that they are independent of the classifier construction process [13].

In general, sampling methods can be categorized into three groups: undersampling, oversampling and hybrid methods. The undersampling methods focus on eliminating data samples from the majority class, whereas the oversampling methods are used for replicating some data samples or creating new data samples from existing ones in the minority class. The hybrid methods combine both oversampling and undersampling to balance the given dataset. In Galar et al. [13], three types of representative approaches, which have been observed to be superior to other related approaches, are based on a combination of sampling methods with ensemble classifiers using either bagging [5] or boosting [11] algorithms. They are briefly described below.

- UnderBagging (UB): UB is a procedure that uses the undersampling method to randomly eliminate several data samples in the majority class; then, the sampled dataset is used to construct 10 (UB1) or 40 (UB4) bagging-based ensemble classifiers [3,13].
- RUSBoost (RUS1): This approach combines random undersampling (RUS) with the boosting algorithm to create classifier ensembles [27]. During the classifier training stage, the boosting algorithm removes data samples from the majority class in each of the 10 iterations of RUS1, but there is no need to assign new weights to the data samples. Instead, the weights of the remaining data samples are simply normalized in the new (balanced) dataset with respect to the total sum of the weights.
- SMOTEBagging (SBAG4): The synthetic minority oversampling technique (SMOTE) is an oversampling method that creates new minority class examples by interpolating several minority class data samples that lie together in the feature space [7]. The SBAG4 method combines the SMOTE preprocessing algorithm with 40 bagging-based ensemble classifiers [30].

2.3. Clustering analysis

Clustering analysis is an unsupervised machine learning technique that is used to group a set of data samples so that those in the same group (i.e., cluster) are more similar to each other than to those in other groups [17]. *K*-means clustering is one of the most widely used clustering algorithms. The aim is to partition n data samples into k clusters so that each data sample belongs to the cluster with the nearest mean. In particular, the nearest mean, also called the cluster center (or centroid), serves as the prototype of the cluster and can be used to represent all data samples in the cluster [15].

Recently, a novel clustering algorithm called affinity propagation has been proposed, which can overcome the limitations of k-means clustering where the number of clusters must be determined before executing the algorithm [12]. The affinity propagation algorithm measures the Euclidean distances between all of the data points to identify their similarities. Then, the message-passing concept is used between data point i and candidate cluster center k depending upon responsibility and availability. Responsibility means how well-suited candidate cluster center k is to serve as the best cluster center (i.e., exemplar) compared to other cluster centers for data point i, whereas availability refers to how appropriate candidate cluster center k is for use as the exemplar for data point i. This process goes through a number of iterations until the best exemplars with the highest quality for all of the data points have been identified.

2.4. Instance selection

Instance selection is an important data preprocessing step in data mining, where the aim is to filter out noisy (or unrepresentative) data samples from a given dataset [14]. The hypothesis is that not all data in the collected dataset used to train a classifier may be equally informative, and the dataset usually contains some noisy data (or outliers). More specifically, classifiers trained using a dataset on which instance selection has been performed are likely to perform better than ones trained using the original dataset [32].

Instance selection can be defined as follows. Given a multiclass training dataset *TA* containing M instances, let X_i be an instance of *TA* where $X_i = (X_{i1}, \mathrm{X}_{\mathrm{i}2}, \ldots, X_{im}, X_{ic})$, meaning that X_i is represented by m-dimensional features and belongs to class c given by X_{ic}. After performing instance selection, a subset of selected samples S is produced, where $S \subseteq \mathrm{TA}$ and $S < M$. The existing instance selection algorithms are all designed to select a subset of the original (multiclass) dataset. They cannot directly be used to select several data samples from only one class dataset, i.e., the majority class for the undersampling purpose as proposed in this paper.

3. The proposed CBIS approach

Fig. 1 shows a block diagram of the proposed cluster-based instance selection (CBIS) approach for undersampling class-imbalanced datasets. It comprises two steps. For instance, let us examine a two-class classification problem, given a two-class (training) dataset D that contains majority and minority class datasets denoted by $D_{majority}$ and $D_{\min ority}$, respectively.

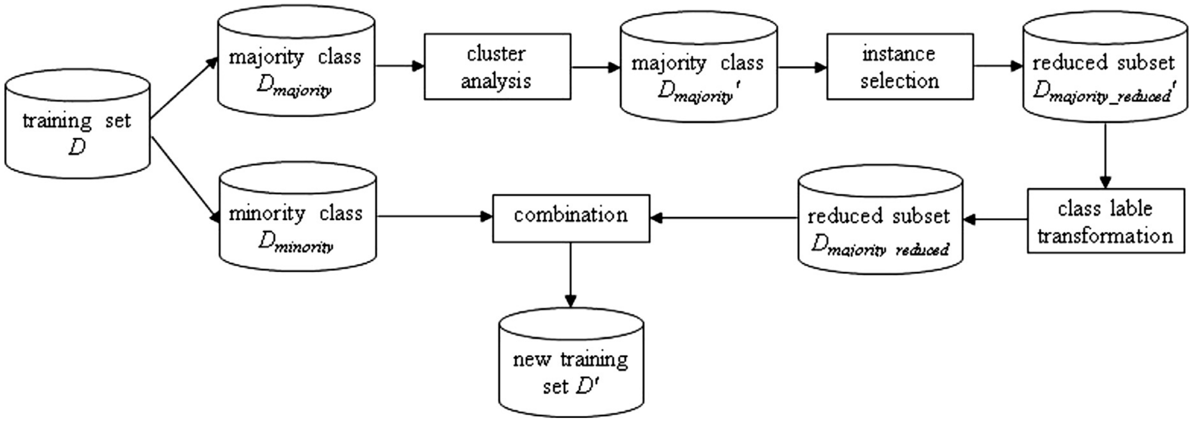


Fig. 1. The steps of the proposed CBIS approach.

The first step is based on using the clustering analysis algorithm to group similar data samples of $D_{majority}$ into a number of G groups (i.e., clusters). In this study, the affinity propagation (AP) algorithm, which does not require predetermining the number of clusters, is used to perform this task. Note that the k-means algorithm will also be used for comparison where the clustering result of AP is the guideline for deciding the value of k.

According to the clustering result, as each grouped data sample is associated with one specific cluster out of G, each of the identified clusters can be regarded as a 'pseudo' subclass of $D_{majority}$. Therefore, a specific class label (i.e., the cluster ID) is assigned to its corresponding data samples. As a result, a new majority class dataset is produced, denoted by $D_{majority}{}'$. The only difference between $D_{majority}$ and $D_{majority}{}'$ involves the class label for each data sample, whereby $D_{majority}$ contains only one class label, while $D_{majority}{}'$ contains G different class labels.

The second step is to perform the instance selection process with $D_{majority}{}'$. In this paper, three different well-known algorithms - the genetic algorithm [6], IB3 [2] and DROP3 [32] – are used individually for comparison. The instance selection algorithm is able to filter out some data samples from each class of $D_{majority}{}'$. The resulting reduced subset of $D_{majority}{}'$, denoted by $D_{majority\_reduced}{}'$, contains a smaller number of data samples than does $D_{majority}{}'$ (as well as $D_{majority}$). Next, the G class labels of $D_{majority\_reduced}{}'$ are transformed into the original class label of the majority class dataset, resulting in a reduced majority class dataset denoted by $D_{majority\_reduced}$.

Finally, $D_{majority\_reduced}$ is combined with $D_{\min ority}$ to produce a new two-class training dataset (D'). It should be noted that in D', the imbalance ratio (IR) between the majority and minority classes is definitely smaller than the IR of D, but this does not necessarily mean that the IR of D' will be 1:1. That is, the reduced number of data samples in the majority class is dependent on the instance selection algorithm used.

Fig. 2 shows the pseudocode of CBIS.

4. Experiments

4.1. Experimental setup

In this paper, 44 two-class imbalanced datasets from the KEEL dataset repository are used for the experiments [13]. The imbalance ratios of these datasets are between 1.8 and 129 with the number of features and data samples ranging from 4 to 20, and 130 to 5500, respectively. Each dataset is divided into five training and testing subsets based on fivefold cross-validation.

The top five best-performing approaches obtained from Galar et al. [13], who compared 37 related approaches, are used as the baselines for performance comparisons. They are UB1, UB4, UB24, RUS1 and SBAG4. In addition, the clustering-based undersampling (CBU) approach proposed recently [20] is also included in the comparison.

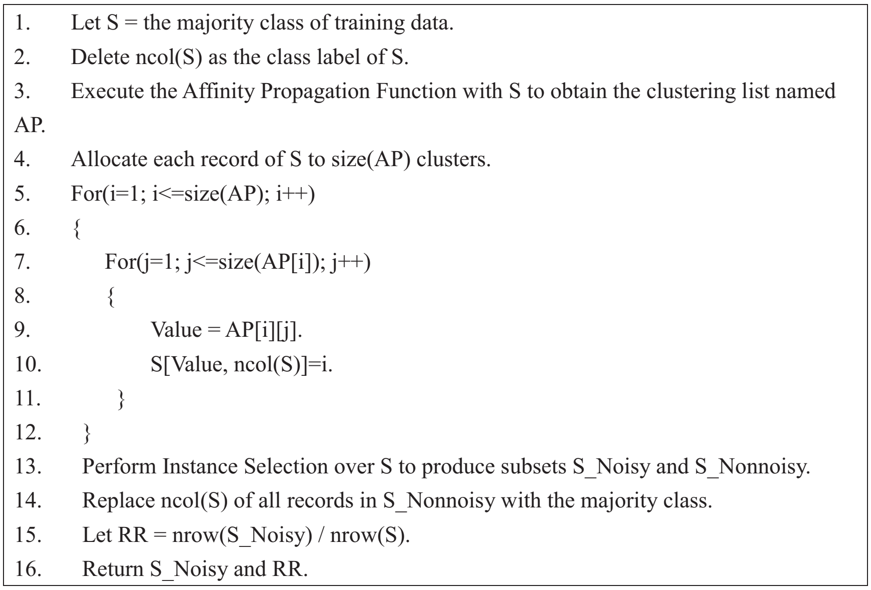
For the proposed CBIS approach, the AP clustering algorithm is combined individually with each of three instance selection algorithms: the genetic algorithm (GA), IB3 and DROP3. Furthermore, two different kinds of C4.5 decision tree ensemble classifiers based on the bagging and boosting algorithms are developed. Finally, the performance of classifiers is determined by examining the area under the curve (AUC) based on the receiver operating characteristic (ROC) curve [10].

4.2. Experimental results

Table 1 lists the performance measures obtained using the six baseline approaches and the proposed CBIS approach applied to 44 datasets. The results show that the proposed CBIS approach mostly outperforms the six baseline approaches regardless of the type or combination of techniques used. In particular, the best baseline, i.e., CBU, performs very similarly

Table 1
Classification results of various approaches.

Dataset	State-of-the-art approaches						CBIS					
							C4.5 boosting ensembles			*C4.5 bagging ensembles*		
	UB1	UB4	UB24	RUS1	SBAG4	CBU	AP + GA	AP + IB3	AP + DROP3	AP + GA	AP + IB3	AP + DROP3
Abalone9-18	0.71	0.719	0.71	0.693	0.745	0.831	0.873	0.849	0.831	0.852	0.894	0.825
Abalone19	0.695	0.721	0.68	0.631	0.572	0.728	0.608	0.624	0.617	0.684	0.728	0.609
Ecoli-0_vs_1	0.969	0.98	0.98	0.969	0.983	0.982	0.958	0.975	0.964	0.973	0.982	0.982
Ecoli-0-1-3-7_vs_2-6	0.726	0.745	0.781	0.794	0.828	0.804	0.888	0.877	0.862	0.887	0.879	0.873
Ecoli1	0.898	0.9	0.902	0.883	0.9	0.927	0.921	0.958	0.959	0.940	0.957	0.952
Ecoli2	0.87	0.884	0.881	0.899	0.888	0.947	0.938	0.942	0.958	0.958	0.934	0.952
Ecoli3	0.882	0.908	0.894	0.856	0.885	0.926	0.898	0.937	0.914	0.940	0.933	0.940
Ecoli4	0.891	0.888	0.899	0.942	0.933	0.95	0.941	0.970	0.966	0.938	0.964	0.960
Glass0	0.818	0.814	0.824	0.813	0.839	0.873	0.811	0.888	0.873	0.863	0.885	0.884
Glass0123vs456	0.894	0.904	0.917	0.93	0.946	0.97	0.893	0.980	0.963	0.963	0.966	0.967
Glass016vs2	0.636	0.754	0.625	0.617	0.559	0.79	0.678	0.775	0.661	0.702	0.713	0.792
Glass016vs5	0.943	0.943	0.943	0.989	0.866	0.964	0.974	0.894	0.994	0.971	0.987	0.994
Glass1	0.748	0.737	0.752	0.763	0.728	0.824	0.788	0.812	0.849	0.781	0.847	0.829
Glass2	0.758	0.769	0.706	0.78	0.779	0.76	0.743	0.741	0.667	0.762	0.766	0.742
Glass4	0.853	0.846	0.871	0.915	0.874	0.853	0.939	0.944	0.976	0.913	0.971	0.968
Glass5	0.949	0.949	0.949	0.943	0.878	0.949	0.976	0.994	0.998	0.935	0.994	0.996
Glass6	0.885	0.904	0.926	0.918	0.931	0.905	0.934	0.951	0.957	0.936	0.934	0.951
Haberman	0.658	0.664	0.668	0.655	0.656	0.603	0.619	0.646	0.611	0.642	0.648	0.674
Iris0	0.99	0.99	0.98	0.99	0.98	0.99	0.970	0.990	0.990	1.000	0.990	0.990
New-thyroid1	0.955	0.964	0.969	0.958	0.975	0.973	0.973	0.979	0.996	0.981	0.997	0.997
New-thyroid2	0.947	0.958	0.938	0.938	0.961	0.924	0.930	0.976	0.980	0.950	0.994	0.994
Page-blocks0	0.952	0.958	0.959	0.948	0.953	0.986	0.987	0.987	0.984	0.990	0.987	0.991
Page-blocks13vs2	0.975	0.978	0.975	0.987	0.988	0.992	0.978	0.998	0.998	0.996	0.997	1.000
Pima	0.758	0.76	0.753	0.726	0.751	0.758	0.756	0.771	0.757	0.795	0.805	0.812
Segmemt0	0.985	0.988	0.986	0.993	0.994	0.996	1.000	0.999	0.998	0.997	0.993	0.996
Shuttle0vs4	1.000	1.000	1.000	1.000	1.000	1.000	0.999	1.000	1.000	0.999	1.000	1.000
Shuttle2vs4	0.988	1.000	1.000	1.000	1.000	0.988	0.996	1.000	0.942	1.000	1.000	1.000
Vehicle0	0.945	0.952	0.954	0.958	0.965	0.99	0.987	0.994	0.991	0.983	0.992	0.984
Vehicle1	0.765	0.787	0.761	0.747	0.769	0.832	0.816	0.817	0.836	0.828	0.825	0.828
Vehicle2	0.957	0.964	0.964	0.97	0.966	0.995	0.995	0.993	0.998	0.990	0.983	0.989
Vehicle3	0.764	0.802	0.784	0.765	0.763	0.827	0.830	0.838	0.835	0.841	0.826	0.839
Vowel0	0.944	0.947	0.9467	0.943	0.988	0.987	0.995	0.972	0.970	0.973	0.981	0.972
Wisconsin	0.957	0.96	0.971	0.964	0.96	0.99	0.984	0.989	0.989	0.983	0.991	0.986
Yeast05679vs4	0.782	0.794	0.814	0.803	0.818	0.869	0.849	0.879	0.886	0.860	0.880	0.841
Yeast1	0.716	0.722	0.721	0.719	0.734	0.747	0.761	0.761	0.744	0.784	0.789	0.790
Yeast1vs7	0.747	0.786	0.773	0.715	0.697	0.768	0.765	0.818	0.791	0.803	0.775	0.814
Yeast1289vs7	0.675	0.734	0.689	0.721	0.658	0.692	0.742	0.777	0.725	0.763	0.605	0.640
Yeast1458vs7	0.563	0.606	0.617	0.567	0.623	0.627	0.708	0.630	0.711	0.670	0.638	0.645
Yeast2vs4	0.94	0.936	0.929	0.933	0.897	0.977	0.980	0.973	0.980	0.980	0.980	0.985
Yeast2vs8	0.761	0.783	0.747	0.789	0.784	0.868	0.879	0.906	0.794	0.888	0.827	0.858
Yeast3	0.94	0.934	0.944	0.925	0.944	0.967	0.958	0.953	0.945	0.970	0.969	0.961
Yeast4	0.86	0.855	0.854	0.812	0.773	0.874	0.878	0.857	0.873	0.913	0.914	0.842
Yeast5	0.964	0.952	0.956	0.959	0.962	0.987	0.981	0.967	0.975	0.979	0.970	0.938
Yeast6	0.864	0.869	0.878	0.823	0.836	0.909	0.865	0.881	0.877	0.892	0.884	0.844
Avg.	0.852	0.864	0.858	0.856	0.853	0.889	0.885	0.897	0.891	0.897	0.899	0.896



```
Let S = the majority class of training data.
Delete ncol(S) as the class label of S.
Execute the Affinity Propagation Function with S to obtain the clustering list named AP.
Allocate each record of S to size(AP) clusters.
For(i=1; i<=size(AP); i++)
{
    For(j=1; j<=size(AP[i]); j++)
    {
        Value = AP[i][j].
        S[Value, ncol(S)]=i.
    }
}
Perform Instance Selection over S to produce subsets S_Noisy and S_Nonnoisy.
Replace ncol(S) of all records in S_Nonnoisy with the majority class.
Let RR = nrow(S_Noisy) / nrow(S).
Return S_Noisy and RR.
```

Fig. 2. Pseudocode of CBIS.

Table 2
The computational costs of GA, IB3 and DROP3.

Algorithm	Processing time (seconds)
IB3	3875.55
DROP3	8132.02
GA	14,138.9

Table 3
The CBIS approach combined with various classifiers.

		C4.5	k-NN	NB	MLP	HC
Boosting ensembles	*AP + GA*	0.885	0.857	0.873	0.898	0.883
	AP + IB3	0.897	0.841	0.867	0.889	0.892
	AP + DROP3	0.891	0.846	0.866	0.889	0.887
Avg.		0.891 (2)	0.848 (5)	0.869 (4)	0.892 (1)	0.887 (3)
Bagging ensembles	*AP + GA*	0.897	0.892	0.878	0.92	0.883
	AP + IB3	0.899	0.888	0.874	0.923	0.892
	AP + DROP3	0.896	0.884	0.873	0.927	0.887
Avg.		0.897 (2)	0.888 (3)	0.875 (5)	0.923 (1)	0.887 (4)

to the worst combination of the proposed approach, i.e., AP + GA using the C4.5 boosting ensemble, where their average performance measures are 0.889 and 0.885, respectively. For the proposed CBIS approach, the best performance is obtained with AP + IB3 combined with the C4.5 bagging-based ensemble classifiers, which significantly outperforms the six baseline approaches ($p < 0.05$).[1]

Since the differences in the performance of various instance selection algorithms are not large (i.e., less than 0.012% and 0.003% for boosting and bagging ensembles, respectively), and there is no significant difference, the computational costs of these three algorithms are further examined to identify the algorithm that is optimal in terms of both effectiveness and efficiency. Table 2 shows the average processing times required to run GA, IB3 and DROP3 algorithms on 44 datasets.[2]

Clearly, executing the IB3 algorithm requires the least amount of time for the instance selection task, whereas GA is the most computationally complex algorithm. Therefore, it can be concluded that an approach based on combining AP and IB3 would be the optimal choice, providing the best classification performance and requiring the least processing time.

In contrast, Table 3 shows the performance of the proposed CBIS approach combined with various ensemble classifiers, including C4.5, k-NN ($k = 5$), naïve Bayes (NB) and multilayer perceptron (MLP). Moreover, hamming clustering (HC) is also

[1] The statistical analysis is based on the Wilcoxon rank-sum test for pairwise comparisons [9].

[2] The computing environment is a PC with Intel® Core(TM) i7-2600 CPU with the frequency of 3.40GHz, and 12GB of RAM.

Table 4
Classification results obtained by combining k-means and instance selection.

k-means	C4.5 boosting ensemble	C4.5 bagging ensemble
$K = 5$	0.889	0.899
$K = 10$	0.89	0.897
$K = 15$	0.895	0.9
$K = 20$	0.895	0.897
$K = 25$	0.894	0.894

Table 5
Classification results for various ensemble classifiers.

	k-NN		NB		MLP	
	Boosting ensembles	*Bagging ensembles*	*Boosting ensembles*	*Bagging ensembles*	*Boosting ensembles*	*Bagging ensembles*
$K = 5$	0.844	0.887	0.866	0.876	0.896	0.922
$K = 10$	0.845	0.885	0.869	0.876	0.895	0.922
$K = 15$	0.846	0.887	0.867	0.875	0.898	0.924
$K = 20$	0.845	0.887	0.866	0.875	0.894	0.924
$K = 25$	0.846	0.884	0.868	0.875	0.898	0.925

used; this method was proposed to achieve the performance comparable to that of related classifiers, such as artificial neural networks and decision trees (Muselli, M. and Liberati, [26]; Muselli, M. and Liberati, [25]).

The results show that the proposed CBIS method combined with the MLP ensemble classifiers has the best performance, whereas the C4.5 ensemble classifiers take the second place. In particular, using the bagging-based MLP ensembles significantly outperforms the other classifiers ($p < 0.05$). The HC method without using the boosting and bagging ensemble techniques can still provide a reasonably good performance compared to that of C4.5 and MLP ensembles.

4.3. Discussion

Originally, the imbalance ratios of 44 datasets were between 1.8 and 129. After performing instance selection with the CBIS approach, the average reduction rates of IB3, DORP3 and GA for 44 majority class datasets are 18%, 16% and 75%, respectively. That is, the GA instance selection algorithm filters out the largest amount of data from the majority class datasets, with an average of 75% of data being removed. Consequently, the imbalance ratios of 44 datasets after performing CBIS based on IB3, DROP3 and GA are between 1.1 and 98.92, 1.44 and 123.1, and 0.1 and 55.72, respectively.

It is interesting to note that although the imbalance ratios obtained with CBIS are still higher than those obtained with the baseline approaches with an imbalance ratio of 1, the final classification performance of CBIS is better than that of the baseline approaches. This indicates that when the majority class dataset contains mostly non-noisy and representative data samples, even if the imbalance ratio is larger than 1, highly effective classifiers can still be constructed.

Finally, to examine the applicability of using other clustering algorithms for the CBIS approach, the *k*-means clustering algorithm is used. In this case, five different numbers of clusters are compared: 5, 10, 15, 20 and 25. Table 4 shows the average classification performance results obtained by combining *k*-means and the three instance algorithms.

As we observe, the differences in performance obtained using different numbers of clusters are very small and, in fact, less than 0.01, which does not represent a significant level of difference. This observation implies that the number of ‘subclasses’ (i.e., clusters) of the majority class does not affect the final instance selection result. Table 5 shows the average classification performance measures of various ensemble classifiers based on various numbers of clusters.

Similar to the results shown in Table 3, it is observed that the CBIS approach combined with the MLP and C4.5 ensemble classifiers results in the best and second-best performance, respectively. They all perform better than the k-NN and NB ensembles with a significant level of difference (i.e., $p < 0.01$). Specifically, the performance measures of the CBIS approach based on AP and *k*-means are very similar and do not have a significant level of difference. Therefore, the above results demonstrate the reliability of the CBIS approach. That is, no matter what types of clustering and instance selection algorithms are combined, the classification performance is better than that of the well-known baseline approaches.

5. Conclusion

In this paper, we introduce a novel undersampling approach called cluster-based instance selection (CBIS). CBIS is composed of two components: clustering analysis and instance selection. The clustering analysis component is used to cluster similar data samples in the majority class dataset into a number of groups that can be regarded as ‘subclasses’ of the majority class. Afterward, the instance selection component is used to filter out unrepresentative data samples from each of the ‘subclasses’.

In the experiment, 44 class-imbalanced datasets are used. In addition, two different clustering algorithms, i.e., affinity propagation and *k*-means algorithms, and three different instance selection algorithms (IB3, DROP3 and GA) are combined individually for performance comparisons. Moreover, different ensemble classifiers using the bagging and boosting algorithms are also constructed to create different combinations of clustering and instance selection algorithms. The experimental results demonstrate the effectiveness of the proposed CBIS approach. In particular, regardless of what kinds of clustering analysis and instance selection algorithms are combined, the MLP ensemble classifiers significantly outperform the six well-known state-of-the-art approaches.

Acknowledgments

The work of the second author was supported in part by the Ministry of Science and Technology of Taiwan under Grant MOST 106-2410-H-182-024, in part by the Featured Areas Research Center Program within the Framework of the Higher Education Sprout Project of theMinistry of Education (MOE) of Taiwan under Grants EMRPD1H0421 and EMRPD1H0551 of the Healthy Aging Research Center, Chang Gung University, and in part by Chang Gung Memorial Hospital, Linkou under Grant NERPD2G0301T. The work of the third author was supported by the Center for Innovative Research on Aging Society from The Featured Areas Research Center Program within the framework of the Higher Education Sprout Project by the Ministry of Education (MOE) in Taiwan.

References

[1] L. Abdi, S. Hashemi, To combat multi-class imbalanced problems by means of over-sampling techniques, IEEE Trans. Knowl. Data Eng. 28 (1) (2016) 238–251.
[2] D.W. Aha, D. Kibler, M.K. Albert, Instance-based learning algorithms, Mach. Learn. 6 (1) (1991) 37–66.
[3] R. Barandela, R.M. Valdovinos, J.S. Sanchez, New applications of ensembles of classifiers, Pattern Anal. Appl. 6 (2003) 245–256.
[4] G.E. Batista, R.C. Prati, M.C. Monard, A study of the behavior of several methods for balancing machine learning training data, ACM SIGKDD Explor. Newsl. 6 (1) (2004) 20–29.
[5] L. Breiman, Bagging predictors, Mach. Learn. 24 (2) (1996) 123–140.
[6] J.R. Cano, F. Herrera, M. Lozano, Using evolutionary algorithms as instance selection for data reduction: an experimental study, IEEE Trans. Evolut. Comput. 7 (6) (2003) 561–575.
[7] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, SMOTE: synthetic minority over-sampling technique, J. Artif. Intell. Res. 16 (2002) 321–357.
[8] N.V. Chawla, N. Japkowicz, A. Kotcz, Editorial: special issue on learning from imbalanced data sets, ACM SIGKDD Explor. Newsl. 6 (1) (2004) 1–6.
[9] J. Demsar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.
[10] T. Fawcett, An introduction to ROC analysis, Pattern Recognit. Lett. 27 (8) (2006) 861–874.
[11] Y. Freund, R.E. Schapire, Experiments with a new boosting algorithm, in: Proceedings of the International Conference on Machine Learning, Bari, Italy, 1996, pp. 148–156. July 3-6.
[12] B.J. Frey, D. Dueck, Clustering by passing messages between data points, Science 315 (5814) (2007) 972–976.
[13] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, F. Herrera, A review on ensembles for class imbalance problem: bagging, boosting and hybrid-based approaches, IEEE Trans. Syst. Man Cybern. – Part C 42 (4) (2012) 463–484.
[14] S. Garcia, J. Derrac, J.R. Cano, F. Herrera, Prototype selection for nearest neighbor classification: taxonomy and empirical study, IEEE Trans. Pattern Anal. Mach. Intell. 34 (3) (2012) 417–435.
[15] J.A. Hartigan, M.A. Wong, Algorithm AS 136: a k-means clustering algorithm, J. R. Stat. Soc. Ser. C 28 (1) (1979) 100–108.
[16] H. He, E.A. Garcia, Learning from imbalanced data, IEEE Trans. Knowl. Data Eng. 21 (9) (2009) 1263–1284.
[17] A.K. Jain, M.N. Murty, P.J. Flynn, Data clustering: a review, ACM Comput. Surv. 31 (3) (1999) 264–323.
[18] K. Kourou, T.P. Exarchos, K.P. Exarchos, M.V. Karamouzis, D.I. Fotiadis, Machine learning applications in cancer prognosis and prediction, Comput. Struct. Biotechnol. J. 13 (2015) 8–17.
[19] B. Krawczyk, M. Galar, L. Jelen, F. Herrera, Evolutionary undersampling boosting for imbalanced classification of breast cancer malignancy, Appl. Soft Comput. 38 (2016) 714–726.
[20] W.-C. Lin, C.-F. Tsai, Y.-H. Hu, J.-S. Jhang, Clustering-based undersampling in class-imbalanced data, Inf. Sci. 409–410 (2017) 17–26.
[21] W.-Y. Lin, Y.-H. Hu, C.-F. Tsai, Machine learning in financial crisis prediction: a survey, IEEE Trans. Syst. Man Cybern. – Part C 42 (4) (2012) 421–436.
[22] V. Lopez, S. del Rio, J.M. Benitez, F. Herrera, Cost-sensitive linguistic fuzzy rule based classification systems under the MapReduce framework for imbalanced big data, Fuzzy Sets Syst. 258 (2015) 5–38.
[23] V. Mahajan, R. Misra, R. Mahajan, Review of data mining techniques for churn prediction in Telecom, J. Inf. Org. Sci. 39 (2) (2015) 183–197.
[24] R. Malhotra, A systematic review of machine learning techniques for software fault prediction, Appl. Soft Comput. 27 (2015) 504–518.
[25] M. Muselli, D. Liberati, Binary rule generation via hamming clustering, IEEE Trans. Knowl. Data Eng. 14 (6) (2002) 1258–1268.
[26] M. Muselli, D. Liberati, Training digital circuits with hamming clustering, IEEE Trans. Circt. Syst. – I 47 (4) (2000) 513–527.
[27] C. Seiffert, T. Khoshgoftaar, J. Van Hulse, A. Napolitano, Rusboost: a hybrid approach to alleviating class imbalance, IEEE Trans. Syst. May Cybern. – Part A 40 (1) (2010) 185–197.
[28] Y. Sun, A.K. Wong, M.S. Kamel, Classification of imbalanced data: a review, Int. J. Pattern Recognit. Artif. Intell. 23 (4) (2009) 687–719.
[29] V. Vigneron, H. Chen, A multi-scale seriation algorithm for clustering sparse imbalanced data: application to spike sorting, Pattern Anal. Appl. 19 (2016) 885–903.
[30] S. Wang, X. Yao, Diversity analysis on imbalanced data sets by using ensemble models, IEEE Symp. Comput. Intell. Data Min. (2009) 324–331.
[31] J. West, M. Bhattacharya, Intelligent financial fraud detection: a comprehensive review, Comput. Secur. 57 (2016) 47–66.
[32] D.R. Wilson, T.R. Martinez, Reduction techniques for instance-based learning algorithms, Mach. Learn. 38 (3) (2000) 257–286.
[33] S.-J. Yen, Y.-S. Lee, Cluster-based under-sampling approaches for imbalanced data distributions, Expert Syst. Appl. 36 (3) (2009) 5718–5727.
[34] H. Yu, C. Mu, C. Sun, W. Yang, X. Yang, X. Zuo, Support vector machine-based optimized decision threshold adjustment strategy for classifying imbalanced data, Knowl.-Based Syst. 76 (2015) 67–78.
[35] S. Zafeiriou, C. Zhang, Z. Zhang, A survey on face detection in the wild: past, present and future, Comput. Vis. Image Understand. 138 (2015) 1–24.
[36] X. Zhang, Y. Li, R. Kotagiri, L. Wu, Z. Tari, M. Cheriet, KRNN: k rare-class nearest neighbour classification, Pattern Recognit. 62 (2017) 33–44.