Information Sciences 409–410 (2017) 17–26

Contents lists available at ScienceDirect

Information Sciences

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

Clustering-based undersampling in class-imbalanced data

Wei-Chao Lin[a], Chih-Fong Tsai[b,*], Ya-Han Hu[c], Jing-Shang Jhang[b]

[a] Department of Computer Science and Information Engineering, Asia University, Taiwan
[b] Department of Information Management, National Central University, Taiwan
[c] Department of Information Management, National Chung Cheng University, Taiwan

ARTICLE INFO

Article history:
Received 15 July 2016
Revised 5 May 2017
Accepted 6 May 2017
Available online 8 May 2017

Keywords:
Class imbalance
Imbalanced data
Machine learning
Clustering
Classifier ensembles

ABSTRACT

Class imbalance is often a problem in various real-world data sets, where one class (i.e. the minority class) contains a small number of data points and the other (i.e. the majority class) contains a large number of data points. It is notably difficult to develop an effective model using current data mining and machine learning algorithms without considering data preprocessing to balance the imbalanced data sets. Random undersampling and oversampling have been used in numerous studies to ensure that the different classes contain the same number of data points. A classifier ensemble (i.e. a structure containing several classifiers) can be trained on several different balanced data sets for later classification purposes. In this paper, we introduce two undersampling strategies in which a clustering technique is used during the data preprocessing step. Specifically, the number of clusters in the majority class is set to be equal to the number of data points in the minority class. The first strategy uses the cluster centers to represent the majority class, whereas the second strategy uses the nearest neighbors of the cluster centers. A further study was conducted to examine the effect on performance of the addition or deletion of 5 to 10 cluster centers in the majority class. The experimental results obtained using 44 small-scale and 2 large-scale data sets revealed that the clustering-based undersampling approach with the second strategy outperformed five state-of-the-art approaches. Specifically, this approach combined with a single multilayer perceptron classifier and C4.5 decision tree classifier ensembles delivered optimal performance over both small- and large-scale data sets.

1. Introduction

In data mining and machine learning, it is difficult to train an effective learning model if the class distribution in a given training data set is imbalanced. This is known as the class imbalance problem. One class might be represented by a large number of examples, whereas the other might be represented by only a few. In addition, for most data mining algorithms, rare objects are much more difficult to identify than common objects [3,33,37]. This is a problem encountered in numerous real-world applications such as medical diagnosis, financial crisis prediction, and e-mail filtering [12]. Furthermore, the primary class of interest in the data mining task is usually the minority (or rare) class.

Without consideration of the class imbalance problem, learning algorithms or constructed models can be overwhelmed by the majority class and can ignore the minority class. For example, consider a two-class data set with an imbalance ratio of 99%, where the majority class constitutes 99% of the data set and the minority class contains only 1%. To minimize the

* Corresponding author.
E-mail address: cftsai@mgt.ncu.edu.tw (C.-F. Tsai).

http://dx.doi.org/10.1016/j.ins.2017.05.008
0020-0255/

error rate, the learning algorithm classifies all of the examples into the majority class, which yields an error rate of 1%. In this case, all of the examples belonging to the minority class are paramount and must be identified as incorrectly classified [24].

Bankruptcy prediction is a practical class imbalance problem [35,39]. In particular, the numbers of bankruptcy cases (i.e. the minority class) are usually much smaller than those of nonbankruptcy cases (i.e. the majority class). The type I error rate, which means that a prediction model incorrectly classifies the bankruptcy case into the nonbankruptcy class, is more critical than the average rate of classification accuracy. This is because higher type I error rates are likely to increase bad debts for financial institutions.

A variety of methods have been proposed to solve this problem. Such methods can be divided into four types: algorithmic-level methods, data-level methods, cost-sensitive methods, and ensembles of classifiers [4,12] (cf. Sections 2.2 and 2.3). In particular, the data-level methods, which focus on preprocessing the imbalanced data sets before constructing the classifiers, are widely considered in the literature. This is because the data preprocessing and classifier training tasks can be performed independently. In addition, according to Galar et al. [12], who conducted a comparative study of numerous well-known approaches, combinations of data preprocessing methods with classifier ensembles perform better than other methods.

Data preprocessing methods are based on resampling the imbalanced training data set before the model training stage. To create balance, the original imbalanced data set can be resampled by oversampling the minority class [8,15] and/or undersampling the majority class [17,23]. Some representative approaches combine oversampling and undersampling data preprocessing with classifier ensembles through boosting [31] or bagging [6] techniques; for example SMOTEBoost [9], RUSBoost [32], OverBagging [36], and UnderBagging [2].

Most of these approaches perform several rounds of random resampling for the majority (i.e. undersampling) or minority (i.e. oversampling) class. In the next group of methods, different balanced training sets are used to train a number of specific classifiers for later combination as classifier ensembles. Of these two resampling strategies, undersampling has been shown to be a better choice than oversampling [5,12]. This is because the oversampling strategy may increase the likelihood of overfitting in the model construction process. However, with the undersampling strategy, some useful data present in the majority class might be eliminated [34].

To overcome the limitations of undersampling, we propose replacing the random undersampling strategy with a clustering technique. The aim of clustering analysis is to group similar objects (i.e. data samples) into the same clusters; the objects in different clusters are different in terms of their feature representations [16]. Therefore, using clustering analysis to undersample the majority class generates a number of clusters, with each cluster containing similar data. Specifically, each cluster centroid (or center), which is based on the mean of similar data in the same group calculated by the k-means algorithm [14], can be used to represent the data in the whole group. In other words, the original data in the same groups are replaced by the cluster centers, thereby reducing the size of the majority class.

In this paper, we demonstrate that this type of clustering-based undersampling strategy can reduce the risk of removing useful data from the majority class, enabling the constructed classifiers (including both single classifiers and classifier ensembles) to outperform classifiers developed using a random undersampling strategy.

The contribution of this paper is twofold. First, we present two strategies of using the *k*-means clustering technique for undersampling in the class imbalance domain problem, which has never been done before. Second, several combinations of the clustering-based undersampling approach with different classification techniques, including five single classifiers and five classifier ensembles, are compared over a large number data sets to identify the optimal solution.

The remainder of this paper is organized as follows. Section 2 overviews the class imbalance problem and some widely used representative approaches that have been compared in the literature. Section 3 describes the research methodology including the clustering-based undersampling method and model construction. Section 4 presents the experimental results, and Section 5 concludes the paper.

2. Literature review

2.1. The class imbalance problem

Class imbalance (or imbalanced classification) is a problem in data sets with skewed distributions of data points. This has the following characteristics [7,12].

- Class overlapping [3]: When the data samples belonging to different classes overlap (as shown in Fig. 1), classifiers have difficulty effectively distinguishing between different classes. In most cases, instances belonging to the minority class are classified into the majority class.
- Small sample size: In practice, collecting sufficient data for class imbalanced data sets is challenging. One solution is to balance the imbalance ratios of the data sets to reduce the misclassification error.
- Small disjuncts: The data samples in the minority classes are distributed in numerous feature spaces, as shown in Fig. 2. This causes a high degree of complication during the classification stage.

Due to a significant difference between the sample sizes of two different classes (i.e. high imbalance ratios), classifiers may treat some of the data points in the minority class as outliers, which produces a very high misclassification error rate

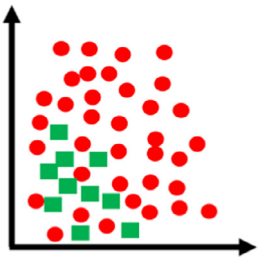

Fig. 1. Example of class overlapping.

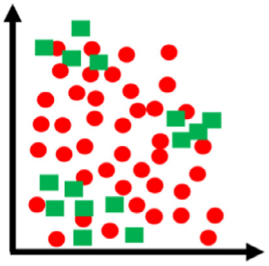

Fig. 2. Example of small disjuncts.

for the minority class. As data set sizes become increasingly larger in numerous real-world applications such as medical decision-making [27], fault diagnosis [40], and face recognition [25], the consequences of class imbalance problems become greater.

2.2. Solutions for data sets with class imbalance

In general, three types of approaches can solve the class imbalance problem. These solutions are based on data, algorithms, or cost sensitivity [22]. Recently, ensembles of classifiers have been employed to handle imbalanced data sets with data-based approaches [4,12]

2.2.1. Data-level solutions

The Data-level solutions are based on preprocessing (or balancing) the collected imbalanced training data set by either undersampling or oversampling strategies. The undersampling approaches are used to reduce the data samples in the majority class, whereas the oversampling approaches are used to increase the data samples in the minority class.

The advantage of these approaches is to make the sampling and classifier training processes independent. Therefore, different sampling approaches can be easily combined with different classifiers. Batista et al. [3] showed that the sampling solutions can effectively solve the class imbalance problem and optimize classifier performance. In addition, Galar et al. [12] reported that combinations of the sampling approaches and different classifier ensembles have been widely considered for the class imbalance problem (cf. Section 2.3).

2.2.2. Algorithm-level solutions

The algorithm-level solutions involve proposing novel algorithms or modifying existing algorithms to directly handle data sets with class imbalance; such algorithms can outperform previously existing algorithms. The threshold method and one-class learning method are widely used approaches in the literature. The threshold method involves setting different threshold values for different classes during the classifier learning stage [37], whereas the one-class learning method entails training the classifier from a training set that contains only one specific class [10,30].

Other types of algorithms, such as evolving clustering in neurofuzzy systems, evolving clustering of dynamic data in spiking neural networks, clustering personalized modeling, and clustering through quantum-inspired evolutionary algorithms, have also been developed to deal with imbalanced data [18–20,28]

2.2.3. Cost-sensitive solutions

The cost-sensitive solutions focus on defining different misclassification costs of classifiers for different classes. Then, a confusion matrix for the misclassification cost can be produced, as shown in Fig. 3.

In Fig. 3, the cost for correct classification is 0; otherwise, if the data sample whose true class is j is incorrectly classified into the i class, its misclassification cost is λ_{ij}. Therefore, according to Eq. (1), one can define the risk of α_i to minimize the

		Prediction	
		Class i	Class j
True	Class i	0	λ_{ij}
	Class j	λ_{ji}	0

Fig. 3. Confusion matrix for the misclassification cost.

misclassification cost [26].

$$R(a_i|x) = \sum_i \lambda_{ij} P(v_j|x) \tag{1}$$

2.3. Combinations of resampling and classifier ensembles

A number of well-known approaches combining resampling techniques and classifier ensembles have been published in the literature, as described in Section 1. Many of such approaches have been used as a baseline for comparison with new proposed approaches [1,13,29,38]. SMOTEBoost, RUSBoost, and UnderBagging have been found to outperform other methods [12]. They are described as follows.

2.3.1. SMOTE

SMOTE (synthetic minority oversampling technique) [9] is one of the most commonly used approaches to counter the imbalance problem. SMOTE is an oversampling approach, which is based on creating synthetic training examples for interpolation with the minority class. These synthetic training examples are created by randomly selecting one or more (depending on the amount of oversampling required) of the k-nearest neighbors of the minority class examples.

Various classification techniques can be employed after the oversampling process. According to Galar et al. [12], SMOTEBoost [9] is one of the most widely used approaches; it combines an oversampling approach and a rule-based learner that performs a boosting procedure.

2.3.2. RUS

RUS (random undersampling) performs similarly to SMOTE, but RUS is based on an undersampling process where some examples are removed from the majority class. Seiffert et al. [32] proposed RUSBoost, combining the random undersampling approach with a boosting procedure, as a simpler, faster, and less complex alternative to performing SMOTEBoost during the model training step.

2.3.3. UB

UB (UnderBagging) is basically a combination of a random undersampling process and a bagging procedure. In Barandela et al. [2], the first work to use UB, the majority class was undersampled and then a balanced training data set was used to construct a bagging-based k-nearest neighbor ensemble ($k = 1$).

Galar et al. [12] used different numbers of balanced training sets (obtained by performing the random undersampling process several times) to train classifier ensembles; each balanced training set was used for a specific classifier. Then, the multiple classifiers were combined by using a bagging combination approach. They found that UB4 and UB24 (where the numbers 4 and 24 mean that 4 and 24 classifiers were combined, respectively) returned the optimal results for this UB approach.

3. Clustering-based undersampling

3.1. Procedure

Fig. 4 shows the procedure for clustering-based undersampling. The processes are described as follows. Given a (two-class) imbalanced data set D composed of a majority class and a minority class, the majority and minority classes contain M and N data points, respectively. The first step is to divide this imbalanced data set into training and testing sets based on the k-fold cross validation method [21]. The second step is to divide the training set into a majority class subset and a minority class subset. Next, the clustering-based undersampling method (cf. Section 3.2) is employed to reduce the number of data samples in the majority class. The reduced majority class subset is then combined with the minority class subset, resulting in a balanced training set. Finally, the classifier is trained and tested by the balanced training and testing sets, respectively.

In contrast to the well-known methods described in Section 2.2, random undersampling is performed without considering any machine-learning-based undersampling method. Subsequently, bagging-based classifiers are trained using the balanced data set. In the training step of the bagging-based classifiers, the majority class data set is divided into a number of

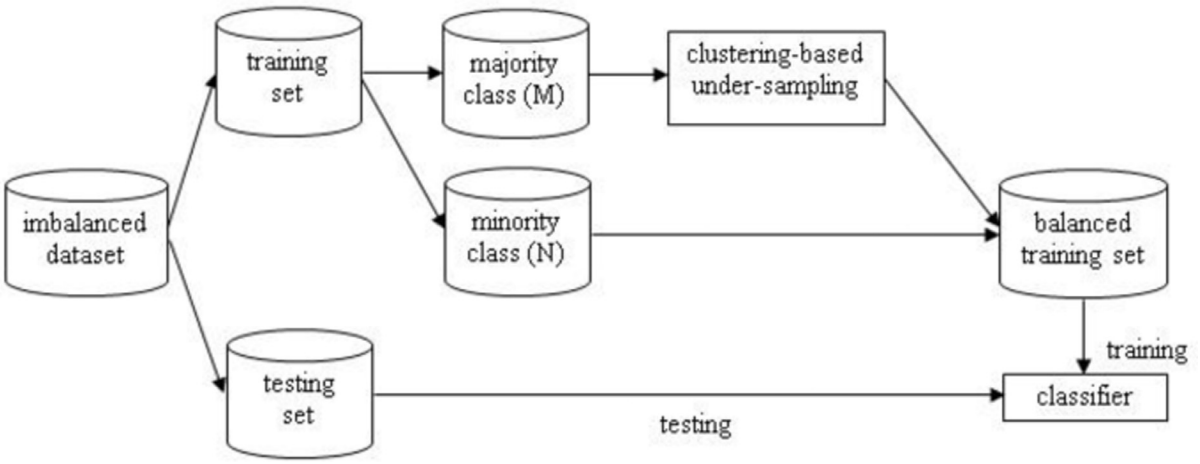

Fig. 4. Clustering-based undersampling procedure.

bags, and random undersampling is performed over each bag. Next, each reduced bag is combined with the minority class data set to train each of the bagging-based classifiers.

3.2. Two strategies for clustering-based undersampling

In this paper, two strategies employing a clustering algorithm to undersample the majority class data set are discussed. Note that although numerous clustering algorithms are mentioned in the literature, we consider only the *k*-means algorithm in this paper because it is widely used and can thus be regarded as a baseline clustering method [16]. The two strategies are described as follows.

In the first strategy, the number of clusters (i.e. k) is set to be equal to the number of data samples in the minority class (i.e. $k = N$). Then, the k cluster centers (or centroids) are produced by the *k*-means algorithm over the M data samples in the majority class. These cluster centers are used to replace the entire majority class data set. Consequently, both the majority and minority class data sets contain the same number of data samples.

In the second strategy, because each cluster center is the mean of the data samples in a cluster, it is a new additional data sample for the majority class. The nearest neighbor of each cluster center, which is a real data sample of M, is selected to replace the k cluster centers used in the first strategy. In particular, the Euclidean distance is used to measure the level of similarity between the cluster center and the data samples in the same cluster. Therefore, the reduced majority class data set contains the same number of data samples as the minority class. Although both strategies produce the same number of data samples to replace the M data samples in the majority class, the data points of both strategies in the feature space are somewhat different.

A sensitivity study was conducted by using different numbers of k (i.e. $k \pm 5$ and $k \pm 10$, where k is N) with these two strategies. The results can clarify the differences in classification performance obtained by using different numbers of data samples in the majority class data set.

4. Experimental procedure

4.1. Experimental setup

4.1.1. Data sets

This paper discusses two experimental studies. The first one was based on the 44 small-scale data sets used by Galar et al. [12]. The imbalance ratios of these data sets are between 1.8 and 129, with the numbers of collected data samples ranging from 130 to 5500. These are all two-class classification data sets. In the second study, two large-scale data sets from the Knowledge Discovery and Data Mining Cup[1] were used, namely the breast cancer and protein homology prediction data sets, which contain 102,294 and 145,751 data samples, respectively. In addition, their imbalance ratios are approximately 163 and 111, respectively. The data set information is presented in Table 1. For classifier training and testing, all of the data sets were divided into 80% and 20% training and testing data sets through the fivefold cross-validation approach.

[1] http://www.kdd.org/kdd-cup.

Table 1
Data set information.

	Datasets	No. of data samples	No. of features	Imbalance ratio
Small scale datasets				
1	Abalone9-18	731	8	16.68
2	Abalone19	4174	8	128.87
3	Ecoli-0_vs_1	220	7	1.86
4	Ecoli-0-1-3-7_vs_2-6	281	7	39.15
5	Ecoli1	336	7	3.36
6	Ecoli2	336	7	5.46
7	Ecoli3	336	7	8.19
8	Ecoli4	336	7	13.84
9	Glass0	214	9	3.19
10	Glass0123vs456	192	9	10.29
11	Glass016vs2	184	9	19.44
12	Glass016vs5	214	9	1.82
13	Glass1	214	9	10.39
14	Glass2	214	9	15.47
15	Glass4	214	9	22.81
16	Glass5	214	9	22.81
17	Glass6	214	9	6.38
18	Haberman	306	3	2.68
19	Iris0	150	4	2
20	New-thyroid1	215	5	5.14
21	New-thyroid2	215	5	4.92
22	Page-blocks0	5472	10	8.77
23	Page-blocks13vs2	472	10	15.85
24	Pima	768	8	1.9
25	Segmemt0	2308	19	6.01
26	Shuttle0vs4	1829	9	13.87
27	Shuttle2vs4	129	9	20.5
28	Vehicle0	846	18	3.23
29	Vehicle1	846	18	2.52
30	Vehicle2	846	18	2.52
31	Vehicle3	846	18	2.52
32	Vowel0	988	13	10.1
33	Wisconsin	683	9	1.86
34	Yeast05679vs4	528	8	9.35
35	Yeast1	1484	8	2.46
36	Yeast1vs7	459	8	13.87
37	Yeast1289vs7	947	8	30.56
38	Yeast1458vs7	693	8	22.1
39	Yeast2vs4	514	8	9.08
40	Yeast2vs8	482	8	23.1
41	Yeast3	1484	8	8.11
42	Yeast4	1484	8	28.41
43	Yeast5	1484	8	32.78
44	Yeast6	1484	8	39.15
Large scale datasets				
1	Breast cancer[a]	102,294	117	163.19
2	Protein homology prediction[b]	145,751	74	111.46

[a] http://www.kdd.org/kdd-cup/view/kdd-cup-2008
[b] http://www.kdd.org/kdd-cup/view/kdd-cup-2004

4.1.2. Classifiers and baseline approaches

To examine the classification performance by clustering-based undersampling, five different classifiers were constructed, namely C4.5, k-nearest neighbor (k-NN), support vector machine (SVM), naïve Bayes (NB), and multilayer perceptron (MLP). In addition, the AdaBoost algorithm was employed to develop classifier ensembles of C4.5, k-NN, SVM, and NB for further comparison. Note that these classifiers were constructed based on the default parameters used in the Weka software package.

The clustering-based undersampling approach was validated by comparison with five state-of-the-art approaches that have been shown to outperform others, namely UB4 (UnderBagging4), UB24 (UnderBagging24), RUS1 (RUSBoost1), SBAG4 (SMOTEBagging4), and UB1 (UnderBagging1) [12]. A C4.5 baseline classifier without undersampling was also constructed for examination. This classifier is used as the baseline in most related studies.

The evaluation metric was based on the area under the receiver operating characteristic (ROC) curve [11].

Table 2
Classification performance of the different approaches with C4.5.

Dataset	State-of-the-art approaches						Clustering-based undersampling			
	UB4	UB24	RUS1	SBAG4	UB1	C4.5	Centers		Centers_NN	
							C4.5	AdaBoost C4.5	C4.5	AdaBoost C4.5
Abalone9-18	0.719	0.71	0.693	0.745	0.71	0.598	0.699	0.808	0.704	0.831
Abalone19	0.721	0.68	0.631	0.572	0.695	0.5	0.639	0.684	0.648	0.728
Ecoli-0_vs_1	0.98	0.98	0.969	0.983	0.969	0.983	0.983	0.982	0.983	0.982
Ecoli-0-1-3-7_vs_2-6	0.745	0.781	0.794	0.828	0.726	0.748	0.715	0.838	0.726	0.804
Ecoli1	0.9	0.902	0.883	0.9	0.898	0.859	0.895	0.94	0.923	0.927
Ecoli2	0.884	0.881	0.899	0.888	0.87	0.864	0.864	0.956	0.878	0.947
Ecoli3	0.908	0.894	0.856	0.885	0.882	0.728	0.847	0.909	0.9	0.926
Ecoli4	0.888	0.899	0.942	0.933	0.891	0.844	0.905	0.949	0.862	0.95
Glass0	0.814	0.824	0.813	0.839	0.818	0.817	0.772	0.89	0.744	0.873
Glass0123vs456	0.904	0.917	0.93	0.946	0.894	0.916	0.914	0.982	0.902	0.97
Glass016vs2	0.754	0.625	0.617	0.559	0.636	0.594	0.645	0.716	0.708	0.79
Glass016vs5	0.943	0.943	0.989	0.866	0.943	0.894	0.943	0.943	0.943	0.964
Glass1	0.737	0.752	0.763	0.728	0.748	0.74	0.713	0.834	0.647	0.824
Glass2	0.769	0.706	0.78	0.779	0.758	0.719	0.658	0.715	0.756	0.76
Glass4	0.846	0.871	0.915	0.874	0.853	0.754	0.651	0.813	0.803	0.853
Glass5	0.949	0.949	0.943	0.878	0.949	0.898	0.888	0.888	0.949	0.949
Glass6	0.904	0.926	0.918	0.931	0.885	0.813	0.858	0.917	0.847	0.905
Haberman	0.664	0.668	0.655	0.656	0.658	0.576	0.62	0.641	0.595	0.603
Iris0	0.99	0.98	0.99	0.98	0.99	0.99	0.99	0.99	0.99	0.99
New-thyroid1	0.964	0.969	0.958	0.975	0.955	0.914	0.938	0.938	0.947	0.973
New-thyroid2	0.958	0.938	0.938	0.961	0.947	0.937	0.938	0.956	0.924	0.924
Page-blocks0	0.958	0.959	0.948	0.953	0.952	0.922	0.934	0.984	0.958	0.986
Page-blocks13vs2	0.978	0.975	0.987	0.988	0.975	0.998	0.911	0.937	0.992	0.992
Pima	0.76	0.753	0.726	0.751	0.758	0.701	0.753	0.77	0.727	0.758
Segmemt0	0.988	0.986	0.993	0.994	0.985	0.983	0.981	0.995	0.98	0.996
Shuttle0vs4	1	1	1	1	1	0.997	1	1	1	1
Shuttle2vs4	1	1	1	1	0.988	0.95	1	1	0.988	0.988
Vehicle0	0.952	0.954	0.958	0.965	0.945	0.93	0.942	0.974	0.948	0.99
Vehicle1	0.787	0.761	0.747	0.769	0.765	0.672	0.722	0.778	0.703	0.832
Vehicle2	0.964	0.964	0.97	0.966	0.957	0.956	0.942	0.994	0.956	0.995
Vehicle3	0.802	0.784	0.765	0.763	0.764	0.664	0.757	0.848	0.731	0.827
Vowel0	0.947	0.9467	0.943	0.988	0.944	0.971	0.941	0.955	0.91	0.987
Wisconsin	0.96	0.971	0.964	0.96	0.957	0.945	0.945	0.99	0.945	0.99
Yeast05679vs4	0.794	0.814	0.803	0.818	0.782	0.68	0.756	0.802	0.769	0.869
Yeast1	0.722	0.721	0.719	0.734	0.716	0.664	0.741	0.74	0.738	0.747
Yeast1vs7	0.786	0.773	0.715	0.697	0.747	0.628	0.66	0.745	0.704	0.768
Yeast1289vs7	0.734	0.689	0.721	0.658	0.675	0.616	0.632	0.687	0.7	0.692
Yeast1458vs7	0.606	0.617	0.567	0.623	0.563	0.5	0.559	0.615	0.603	0.627
Yeast2vs4	0.936	0.929	0.933	0.897	0.94	0.831	0.914	0.942	0.882	0.977
Yeast2vs8	0.783	0.747	0.789	0.784	0.761	0.525	0.629	0.755	0.778	0.868
Yeast3	0.934	0.944	0.925	0.944	0.94	0.86	0.901	0.958	0.926	0.967
Yeast4	0.855	0.854	0.812	0.773	0.86	0.614	0.722	0.826	0.857	0.874
Yeast5	0.952	0.956	0.959	0.962	0.964	0.883	0.954	0.982	0.96	0.987
Yeast6	0.869	0.878	0.823	0.836	0.864	0.712	0.691	0.841	0.818	0.909
Avg.	0.864	0.858	0.856	0.853	0.852	0.793	0.82	0.873	0.84	0.889

4.2. Study I

Table 2 shows the classification performance obtained with the different approaches using the C4.5 classifier. For the proposed clustering-based undersampling approach, *Centers* refers to the cluster centers (strategy 1) and *Centers_NN* refers to the nearest neighbors of the cluster centers (strategy 2).

As indicated by the performance results, the proposed clustering-based undersampling approaches (i.e. Centers and Centers_NN) combined with AdaBoost C4.5 demonstrated the highest and second-highest classification performance in terms of average classification accuracy. The analysis of variance revealed these two approaches to significantly outperform the others ($p < 0.001$). In addition, using the nearest neighbors of the cluster centers is preferable to using the cluster centers for the proposed approach. Their classification performance were determined to be significantly different ($p < 0.001$).

Table 3 presents the average classification accuracy of different classification techniques for the proposed approach. According to these results, the cluster center strategy induced the single MLP classifier to deliver the optimal tested performance, with 90.5% classification accuracy, which was observed to be significantly different from the others ($p < 0.001$), except for MLP ensembles. Similarly, MLP and MLP ensembles based on the nearest neighbors of the cluster center strategy significantly outperformed the other classifiers ($p < 0.001$). However, only two classifiers, C4.5 and SVM, were shown to benefit from the AdaBoost technique.

Table 3
Average classification accuracy of different classifiers.

	Centers		Centers_NN	
	Single classifier	*Ensemble classifier*	*Single classifier*	*Ensemble classifier*
C4.5	0.82	0.873	0.84	0.889
k-NN	0.884	0.858	0.876	0.852
SVM	0.763	0.782	0.763	0.798
NB	0.864	0.865	0.865	0.868
MLP	0.905	0.903	0.91	0.904

Table 4
Average classification accuracy of different classifiers for different numbers of cluster centers.

		k = N	k = N − 5	k = N − 10	k = N + 5	k = N + 10
C4.5	*Single*	0.814	0.804	0.796	0.812	0.812
	Ensemble	0.867	0.855	0.833	0.864	0.861
k-NN	*Single*	0.856	0.842	0.841	0.85	0.857
	Ensemble	0.856	0.842	0.841	0.85	0.857
SVM	*Single*	0.755	0.697	0.669	0.715	0.696
	Ensemble	0.761	0.741	0.722	0.749	0.756
NB	*Single*	0.854	0.852	0.849	0.851	0.856
	Ensemble	0.854	0.855	0.85	0.853	0.855
MLP	*Single*	0.898	0.897	0.885	0.9	0.895
	Ensemble	0.895	0.895	0.888	0.894	0.896

Finally, we performed a sensitivity analysis on classification performance associated with different numbers of cluster centers (i.e. $k \pm 5$ and $k \pm 10$). Table 4 shows the average classification accuracy of different classifiers using this third strategy. Note that N means the number of data samples in the minority class.

According to these results, when the numbers of data samples in the majority class were less than those in the minority class (i.e. $k = N - 5$ and $N - 10$), the classification performance of C4.5 and k-NN gradually degraded, with the significance level being 0.1. However, when the numbers of data samples in the majority class were slightly higher than those in the minority class (i.e. $k = N + 5$ and $N + 10$), the difference of the classification performance between $k = N$, $N + 5$, and $N + 10$ was very small.

For the SVM classifier, the optimal performance occurred when $k = N$. Regardless of how many data samples in the majority class were added or deleted, $k = N - 5$, $N - 10$, $N + 5$, or $N + 10$, the SVM classifier performed significantly worse than when $k = N$ ($p < 0.05$).

However, the NB and MLP classifiers performed similarly even when the number of data samples in the majority class differed, with the difference in performance being less than 1%, and their performance levels were not significantly different.

4.3. Study II

In the second experimental study, the proposed approach based on the nearest neighbors of the cluster centers (i.e. Centers_NN) combined with AdaBoost was compared with the best state-of-the-art method, UB4, over two large-scale data sets. Figs. 5 and 6 show the results obtained by the different classifier ensembles using the breast cancer and protein homology data sets, respectively. The proposed C4.5 classifier-ensemble-based approach significantly outperformed all other approaches over both large-scale data sets ($p < 0.05$).

Some conclusions can be drawn from these experimental results, to be used as guidelines for future research. For small-scale data sets that contain small imbalance ratios, using the proposed nearest neighbors of the cluster centers (i.e. Centers_NN) combined with a single MLP classifier is preferable. However, for large-scale data sets that contain relatively large imbalance ratios, using the proposed Centers_NN strategy combined with C4.5 classifier ensembles can produce the highest rate of classification accuracy.

5. Conclusion

Most related studies attempting to solve the class imbalance problem have focused on random undersampling and oversampling to balance imbalanced data sets. Classifier ensembles are typically trained with several different balanced data sets for later classification. In this paper, three undersampling strategies based on a clustering technique are introduced. The research objective was to demonstrate the applicability of using a small number of cluster centers and their nearest neighbors to represent all data samples of the majority class.

For the first experimental study conducted on 44 small-scale data sets, with numbers of data samples ranging from 130 to 5500 and imbalance ratios ranging from 1.8 and 129, the clustering-based undersampling approach using the nearest

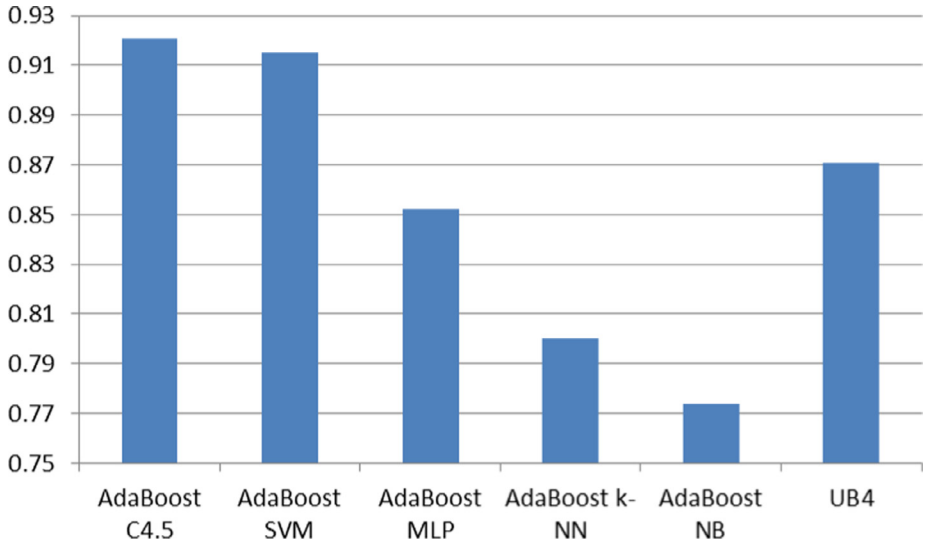

Fig. 5. Classification accuracy of the different classifier ensembles over the breast cancer data set.

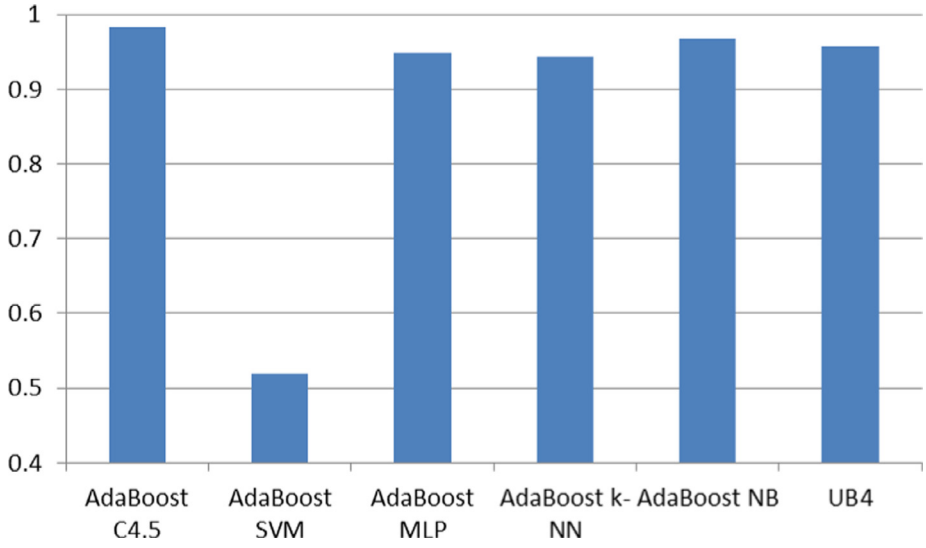

Fig. 6. Classification accuracy of the different classifier ensembles over the protein homology data set.

neighbors of the cluster centers was used; thus, the C4.5 classifier ensembles, single k-NN classifiers, NB classifier ensembles, and single MLP classifier outperformed five other state-of-the-art approaches. In particular, combining this approach with MLP provided the highest rate of classification accuracy.

The second experimental study involved the use of two large-scale data sets, containing over 100,000 data samples and imbalance ratios of 111 and 163. The results show that the clustering-based undersampling approach using the nearest neighbors of the cluster centers was the most preferable choice for undersampling the imbalanced data sets. In addition, combining this approach with C4.5 classifier ensembles outperformed all other tested approaches.

Two issues can be considered in the future. First, because feature selection and instance selection are two crucial data preprocessing tasks in data mining, which are used to filter out unrepresentative features and data samples from a given data set, respectively, it would be very useful to examine the effects of performing feature and instance selection on resampling results. Second, the clustering-based undersampling approach could be combined with novel and/or modified classification algorithms (cf. Section 2.2.2) as algorithm-level solutions for further comparisons.

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] R. Barandela, R.M. Valdovinos, J.S. Sanchez, New applications of ensembles of classifiers, *Pattern Anal. Appl.* 6 (2003) 245–256.
[3] G.E.A.P.A. Batista, R.C. Prati, M.C. Monard, A survey of the behavior of several methods for balancing machine learning training data, *SIGKDD Explor.* 6 (1) (2004) 20–29.
[4] C. Bayan, R. Fisher, Classifying imbalanced data sets using similarity based hierarchical decomposition, *Pattern Recognit.* 48 (2015) 1653–1672.
[5] J. Blaszczynski, J. Stefanowski, Neighborhood sampling in bagging for imbalanced data, Neurocomputing 150 (2015) 529–542.
[6] L. Breiman, Bagging predictors, *Mach. Learn.* 24 (2) (1996) 123–140.
[7] N.V. Chawla, Data mining for imbalanced datasets: an overview, in: O. Maimon, L. Rokach (Eds.), Data Mining and Knowledge Discovery Handbook, Springer-Verlag, New York, 2005, pp. 853–867.
[8] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, SMOTE: synthetic minority over-sampling technique, *J. Artifi. Intell. Res.* 16 (2002) 321–357.
[9] N.V. Chawla, A. Lazarevic, L.O. Hall, K.W. Bowyer, SMOTEBoost: improving prediction of the minority class in boosting, in: European Conference on Principles and Practice of Knowledge Discovery in Databases, 2003, pp. 107–119.
[10] W.W. Cohen, Fast effective rule induction, in: International Conference on Machine Learning, 1995, pp. 115–123.
[11] T. Fawcett, An introduction to ROC analysis, *Pattern Recognit. Lett.* 27 (2006) 861–874.
[12] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, F. Herrera, A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches, *IEEE Trans. Syst. Man Cybern. – Part C* 42 (4) (2012) 463–484.
[13] M. Galar, A. Fernandez, E. Barrenechea, F. Herrera, EUSBoost: enhancing ensembles for highly imbalanced data-sets by evolutionary undersampling, *Pattern Recognit.* 46 (2014) 3460–3471.
[14] 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.
[15] S. Hu, Y. Liang, L. Ma, Y. He, MSMOTE: improving classification performance when training data is imbalanced, in: International Workshop on Computer Science and Engineering, 2009, pp. 13–17.
[16] A.K. Jain, M.N. Murty, P.J. Flynn, Data clustering: a review, *ACM Comput. Surv.* 31 (3) (1999) 264–323.
[17] N. Japkowicz, The class imbalance problem: significance and strategies, in: International Conference on Artificial Intelligence, 2000, pp. 111–117.
[18] N. Kasabov, Evolving connectionist systems for adaptive learning and knowledge discovery: trends and directions, *Knowl.-Based Syst.* 80 (2015) 24–33.
[19] N. Kasabov, M. Doborjeh, Z. Doborjeh, Mapping, learning, visualisation, classification and understanding of fMRI data in the NeuCube spatio temporal data machine, *IEEE Trans. Neural Netw. Learn. Syst.* (2016) Manuscript Number: TNNLS-2016-P-6356, 2016, doi:10.1109/TNNLS.2016.2612890.
[20] N. Kasabov, V. Feigin, Z.-G. Hou, Y. Chen, L. Liang, R. Krishnamurthi, P. Parmar, Evolving spiking neural networks for personalised modelling, classification and prediction of spatio-temporal patterns with a case study on stroke, Neurocomputing 134 (2014) 269–279.
[21] R. Kohavi, A study of cross-validation and bootstrap for accuracy estimation and model selection, in: International Joint Conference on Artificial Intelligence, 2, 1995, pp. 1137–1143.
[22] S. Kotsiantis, D. Kanellopoulos, P. Pintelas, Handling imbalanced datasets: a review, *GESTS Int. Trans. Comput. Sci. Eng.* 30 (1) (2006) 25–36.
[23] M. Kubat, S. Matwin, Addressing the curse of imbalanced training sets: one-sided selection, in: International Conference on Machine Learning, 1997, pp. 179–186.
[24] X.-Y. Liu, J. Wu, Z.-H. Zhou, Exploratory undersampling for class-imbalance learning, *IEEE Trans. Syst. Man Cybern. – Part B* 39 (2) (2009) 539–550.
[25] Y.-H. Liu, Y.-T. Chen, Total margin based adaptive fuzzy support vector machines for multiview face recognition, in: IEEE International Conference on Systems, Man and Cybernetics, 2, 2005, pp. 1704–1711.
[26] R. Longadge, S.S. Dongrre, L. Malik, Class imbalance problem in data mining: review, *Int. J. Comput. Sci. Netw.* 2 (1) (2013) 83–87.
[27] M.A. Mazurowski, P.A. Habas, J.M. Zurada, J.Y. Lo, J.A. Baker, G.D. Tourassi, Training neural network classifiers for medical decision making: the effects of imbalanced datasets on classification performance, *Neural* Netw. 21 (2-3) (2008) 427–436.
[28] A. Mohemmed, S. Schliebs, S. Matsuda, N. Kasabov, SPAN: spike pattern association neuron for learning spatio-temporal sequences, *Int. J. Neural Syst.* 22 (4) (2012) 1–16.
[29] L. Nanni, C. Fantozzi, N. Lazzarini, Coupling different methods for overcoming the class imbalance problem, Neurocomputing 158 (2015) 48–61.
[30] B. Raskutti, A. Kowalczyk, Extreme rebalancing for SVMs: a case study, *ACM SIGKDD Explor.* 6 (1) (2004) 60–69.
[31] R.E. Schapire, The strength of weak learnabilty, *Mach. Learn.* 5 (2) (1990) 197–227.
[32] C. Seiffert, T. Khoshgoftaar, J. Van Hulse, A. Napolitano, RUSBoost: a hybrid approach to alleviating class imbalance, *IEEE Trans. Syst. Man Cybern. – Part A* 40 (1) (2010) 185–197.
[33] Y. Sun, A.K.C. Wong, M.S. Kamel, Classification of imbalanced data: a review, *Int. J. Pattern Recognit. Artif. Intell.* 23 (4) (2009) 687–719.
[34] Z. Sun, Q. Song, X. Zhu, H. Sun, B. Xu, Y. Zhou, A novel ensemble method for classifying imbalanced data, *Pattern Recognit.* 48 (2015) 1623–1637.
[35] K. Taehoon, A. Hyunchul, A hybrid under-sampling approach for better bankruptcy prediction, *J. Intell. Inf. Syst.* 21 (2) (2015) 173–190.
[36] S. Wang, X. Yao, Diversity analysis on imbalanced data sets by using ensemble models, in: IEEE International Symposium on Computational Intelligence and Data Mining, 2009, pp. 324–331.
[37] G.M. Weiss, Mining with rarity: a unifying framework, *SIGKDD Explor.* 6 (1) (2004) 1–7.
[38] X. Zhang, Q. Song, G. Wang, K. Zhang, L. He, X. Jia, A dissimilarity-based imbalance data classification algorithm, *Appl. Intell.* 42 (2015) 544–565.
[39] L. Zhou, Performance of corporate bankruptcy prediction models on imbalanced dataset: the effect of sampling methods, *Knowl.-Based Syst.* 41 (2013) 16–25.
[40] Z.-B. Zhu, Z.-H. Song, Fault diagnosis based on imbalance modified kernel Fisher discriminant analysis, *Chem. Eng. Res. Des.* 88 (8) (2010) 936–951.