Learning from Imbalanced Data Using Ensemble Methods and Cluster-based Undersampling

Parinaz Sobhani[1, *], Herna Viktor[1], Stan Matwin[2]

[1] School of Electrical Engineering and Computer Science, University of Ottawa
{psobh090, hviktor}@uottawa.ca
[2] Faculty of Computer Science, Dalhousie University
stan@cs.dal.ca

Abstract. Imbalanced data, where the number of instances of one class is much higher than the others, are frequent in many domains such as fraud detection, telecommunications management, oil spill detection and text classification. Traditional classifiers do not perform well when considering data that are susceptible to both within-class and between-class imbalances. In this paper, we propose the ClustFirstClass algorithm that employs cluster analysis to aid classifiers when aiming to build accurate models against such imbalanced data sets. In order to work with balanced classes, all minority instances are used together with the same number of majority instances. To further reduce the impact of within-class imbalance, majority instances are clustered into different groups and at least one instance is selected from each cluster. Experimental results demonstrate that our proposed ClustFirstClass algorithm yields promising results compared to the state-of-the art classification approaches, when evaluated against a number of highly imbalanced datasets.

Keywords: Imbalanced data, Undersampling, Ensemble Learning, Cluster analysis

1 Introduction

Learning from data in order to predict class labels has been widely studied in machine learning and data mining domains. Traditional classification algorithms assume balanced class distributions. However, in many applications the number of instances of one class is significantly less than in the other classes. For example, in credit card fraud detection, direct marketing, detecting oil spills from satellite images and network intrusion detection the target class has fewer representatives compared to other classes. Due to the increase of these applications in recent years, learning in the presence of imbalanced data has become an important research topic.

It has been shown that when classes are well separated, regardless of the imbalanced ratio, instances can be correctly classified using standard learning algorithms [1]. However, having class imbalance in complex datasets results in the misclassification of data, especially of the minority class instances. Such data complexity covers issues such as overlapping classes, within-class imbalance, outliers and noise.

Within-class imbalance occurs when a class is scattered into smaller sub-parts representing separate subconcepts [2]. Subconcepts with limited representatives are called "small disjuncts" [2]. Classification algorithms are often not able to learn small disjuncts. This problem is more severe in the case of undersampling techniques. This is due to the fact that the probability of randomly selecting an instance from small disjuncts within the majority class is very low. These regions may thus remain unlearned. The main contribution of this paper is to address this issue by employing clustering techniques.

In this paper, a novel binary-class classification algorithm is suggested to handle data imbalance, mainly within-class and between-class imbalance. Our ClustFirstClass technique employs clustering techniques and ensemble learning methods to address these issues. In order to obtain balanced classes, all minority instances are used together with the same number of majority instances, as obtained after applying a clustering algorithm. That is, to reduce the impact of within-class imbalance majority instances are clustered into different groups and at least one instance is selected from each cluster. In our ClustFirstClass method, several classifiers are trained with the above procedure and combined to produce the final prediction results. By deploying several classifiers rather than a single classifier, information loss due to neglecting part of majority instances is reduced.

The rest of this paper is organized as follows. The next section presents related works for classification of imbalanced data. We detail our ClustFirstClass method in Section 3. Section 4 describes the setup and results of implementing and comparing of our algorithm with other state-of-the-art methods. Finally, Section 5 concludes the paper.

2 Related Work

Imbalanced class distribution may be handled by two main approaches. Firstly, there are sampling techniques that attempt to handle imbalance at data level by resampling original data to provide balanced classes. The second category of algorithms modifies existing classification methods at algorithmic level to be appropriate for imbalanced setting [3]. Most of the previous works in the literature have been concentrated on finding a solution at the data level.

Sampling techniques can improve classification performance in most imbalanced applications [4]. These approaches are broadly categorized as undersampling and oversampling techniques. The main idea behind undersampling techniques is to reduce the number of majority class instances. Oversampling methods, on the other hand, attempt to increase the number of minority examples to have balanced datasets. Both simple under- and oversampling approaches suffer from their own drawbacks. The main drawback of undersampling techniques is information loss due to neglecting part of majority instances. A major drawback of oversampling methods is the risk of overfitting, as a consequence of repeating minority examples many times.

In recent years, using ensemble approaches for imbalanced data classification has drawn lots of interest in the literature. Since ensemble algorithms are naturally designed to improve accuracy, applying them solely on imbalanced data does not solve the problem. However, their combination with other techniques such as under- and oversampling methods has shown promising results [16]. In [13] by integrating bagging with undersampling techniques better results are obtained. In [5], an ensemble algorithm, namely EasyEnsemble, has been introduced to reduce information loss. EasyEnsemble obtains different subsets by independently sampling from majority instances and combines each subset with all the minority instances to train base classifiers of the ensemble learner. In another work that extends bagging ensembles [20], the authors propose the use of so-called roughly balanced (RB) bagging ensembles, where the number of instances from the classes is averaged over all the subsets. A drawback of these bagging approaches is that they choose instances randomly, i.e. without considering the distribution of the data within each class while in [12] it has shown that one of the key factor in the success of ensemble method is majority instance selection strategy.

Cluster-based sampling techniques have been used to improve the classification of imbalanced data. Specifically, they have introduced “an added element of flexibility” that has not been offered by most of previous algorithms [4]. Jo et al. have suggested a cluster-based oversampling method to address both within-class and between-class imbalance [2]. In this algorithm, the K-means clustering algorithm is independently applied on minority and majority instances. Subsequently, each cluster is oversampled such that all clusters of the same class have an equal number of instances and all classes have the same size. The drawback of this algorithm, like most of oversampling algorithms, is the potential of overfitting the training data. In this paper, we also attempt to handle within and between class imbalances by employing clustering techniques. However, in our work we use undersampling techniques instead of oversampling, in order to avoid this drawback. In [6], a set of undersampling methods based on clustering (SBC) is suggested. In their approach, all the training data are clustered in different groups and based on the ratio of majority to minority samples in each cluster, a number of majority instances are selected from each cluster. Finally, all minority instances are combined with selected majority examples to train a classifier. Our approach is completely different as we only cluster majority instances and the same number of majority instances is selected from all clusters.

3 Proposed Algorithm

In this section, a new cluster-based under-sampling approach, called ClustFirstClass, is presented for binary classification. However, it can be easily extended to multiclass scenarios. This method is capable of handling between-class imbalance by having the same number of instances from minority and majority classes and within-class imbalance by focusing on all clusters within a class equally.

To have more intuition why clustering is effective for classification of imbalanced data, consider the given distribution of Figure 1. In this figure, circles represent majority class instances and squares are instances of minority class. Each of these classes contains several subconcepts. In order to have balanced classes, it follows that eight majority instances should be selected and combined with minority representatives to train a classifier. If these instances are randomly chosen, the probability of selecting an instance from region 1 and 2 will be low. Thus, the classifier will have difficulty classifying instances in these regions correctly. In general, the drawback of randomly selecting small number of majority class instances is that small disjuncts with less representative data may remain unlearned. By clustering majority instances in different groups and then selecting at least one instance from each cluster, this problem can be resolved.

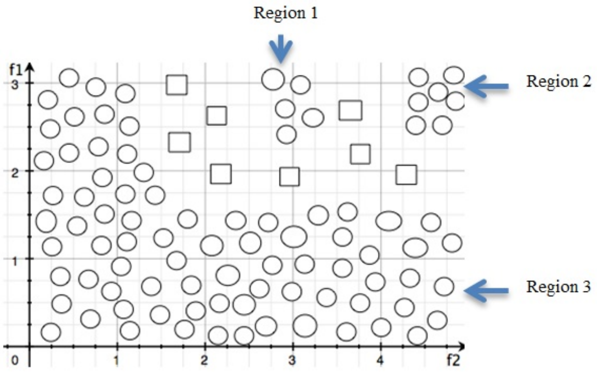


Fig. 1. A dataset with between and within class imbalance

3.1 Under-sampling based on clustering and K-nearest neighbor

In this group of methods, a single classifier is trained using all minority instances and equal number of majority instances. In order to have a representative from all subconcepts of the majority class, these instances are clustered into disjoint groups and one instance is selected from each cluster. However, rather than blindly selecting an instance, we attempt to choose more informative representative from each cluster. Principally, the difference between methods of this group is how these samples are selected from each cluster.

One of the most common representatives of a cluster is the cluster centroid. In our first suggested algorithm, clusters' centroids are combined with minority instances to train a classifier. For the rest of our methods, we follow the same procedures as presented in [7] to choose one instance from each cluster based on K-nearest neighbor (KNN) classifier. These three methods are widely used and have shown to produce good results in many domains [7]. Firstly, NearMiss1 selects the majority example from each cluster that has the minimum average distance to the three closest minority instances, as compared to the other examples in its cluster. In the same way, in NearMiss2, the example with minimum distance to its three farthest minority instances is chosen. The third alternative involves choosing the instance from each cluster that has the "most distance" to its three minority nearest neighbors.

3.2 Under-sampling based on clustering and ensemble learning

The main drawback of most undersampling methods, including those methods suggested earlier in this paper, is the information loss caused by considering a small set of majority instances and subsequently neglecting other majority class instances that may contain useful information for classification. Ensemble methods can solve this problem by using more instances of the majority class in different base learners [4]. In our proposed ensemble method, several classifiers are trained and combined to generate the final results. Each classifier is trained by selecting at least one sample from each cluster. Recall that the advantage of using cluster-based sampling instead of blind sampling is that all subconcepts are represented in training data. Therefore, none of them remains "unlearned".

The proposed ensemble algorithm is developed by training several base classifiers that are combined using a weighted majority voting combination rule, where the weight of each classifier is proportional to inverse of its error on the whole training set. Each learner is trained using *Dmin*, whole minority instances, and *Emaj*, selected majority instances, where *Emaj* contains |*Dmin*|/*k* randomly selected instances from each cluster. By assigning a value between 1 and |*Dmin*| to *k*, a balanced learner is obtained, while ensuring that instances from all subconcepts of majority class participate in training a classifier. The following pseudo-code describes our proposed algorithm in more details.

```
ClusFirstClass Algorithm
Input: D ={(x_i, y_i)}, i=1,…, N
Divide D into $D_{min}$ and $D_{maj}$
Cluster $D_{maj}$ into $k$ partition $P_i$ i=1,…,k
For each classifier $C_j$ j=1,…,m
    For each cluster $P_i$
        $E_{maj}$+= Randomly selected $|Dmin|/k$ instances of $P_i$
    End For
    Tr = $E_{maj}$ + $D_{min}$
    Train $C_j$ using Tr
    $e_j$= Error rate of  $C_j$ on D
    $W_j$= log (1/ $e_j$)
 End For
Output: $C_{final}(x) = \mathrm{argmax}_c \sum_{i=1}^{m} W_i \, |C_i(x) == c|$
```

4 Experiments and Results

In this section, first, common evaluation metrics for imbalanced data are introduced and then datasets and experimental setting that are used in this paper are presented.

Finally, our proposed algorithms are evaluated and compared with several state-of-the-art methods.

4.1 Evaluation Criteria

For imbalanced datasets, it is not sufficient to evaluate the performance of the classifier by only considering the overall accuracy [4]. In this paper, following other researchers, we use the F-measure and G-mean measures to evaluate the performance of different algorithms. Both F-measure and G-mean are functions of confusion matrix, a popular representation of the classifier performance. The F-measure considers both precision and recall at the same time while G-mean combines sensitivity and specificity as an evaluation metric.

4.2 Datasets and Experimental Settings

In this section, first artificial and real datasets for our experiments are introduced and subsequently more details about our experimental settings are described. Our proposed algorithm is particularly effective in presence of within and between class imbalances. To evaluate efficiency of our proposed method, it is applied on two sets of artificial datasets with varying degree of between class imbalances and different number of subconcepts. Furthermore, it is tested on real datasets from UCI repository [9].

Table 1. Description of uni-dimensional artificial datasets

Imbalance ratio	**Dataset Size**	**0-0.25** +		**0.25-50** -		**0.50-0.75** +		**0.75-1** -	
1:9	80	4		68		4		4	
	400	20		340		20		20	
	1600	80		1280		80		80	
1:3	80	10		50		10		10	
	400	50		250		50		50	
	1600	200		1000		200		200	
Imbalance ratio	**Dataset Size**	**0-0.125** +	**0.125-0.25** -	**0.25-0.375** +	**0.375-0.50** -	**0.50-0.675** +	**0.675-0.75** -	**0.75-0.875** +	**0.875-1** -
1:9	80	2	23	2	23	2	3	2	23
	400	10	13	10	119	10	119	10	119
	1600	40	466	40	466	40	466	40	42
1:3	80	5	18	5	6	5	18	5	18
	400	25	27	25	91	25	91	25	91
	1600	100	366	100	366	100	366	100	102

To create artificial datasets with varying degree of imbalance ratio and the number of subconcepts, we follow the same procedure as [1] with one difference that majority class as well as minority class has small disjuncts. In our artificial datasets, majority class instances have at least one small disjuncts. As in [1], three parameters are considered to create different datasets: dataset size, number of subconcepts and the imbalance ratio. Two sets of artificial datasets one uni-dimensional and the other multi-dimensional are generated.

Table 1 describes the number and label of data in each subconcept in uni-dimensional space. Data are distributed uniformly in intervals. For datasets with eight subconcepts, one of the intervals of majority data (negative label) is selected randomly as small disjunct with less representative data. Multi-dimensional datasets have five dimensions and we have the same clusters as [1]. The definition of subconcepts and dataset sizes is the same as described datasets in table 1.

In [8], a benchmark of highly imbalanced datasets from UCI repository is collected and prepared for binary classification task. We selected 8 datasets with wide range of imbalance ratios (from 9 to 130), sizes (from 300 to over 4000 examples) and attributes (purely nominal, purely continuous and mixed) from this benchmark. Table 2 shows the summary of these datasets. Here, all the nominal features have been converted to binary values with multiple dimensions. Following [8], datasets that had more than two classes have been modified by selecting one target class as positive, and considering the rest of the classes as being negative. Continuous features have been normalized to avoid the effect of different scales for different attributes especially for our distance measurements.

Table 2. The Summary of Datasets. In Features, N and C represent Nominal and Continuous respectively.

Dataset	Size	Features	Target	Imbalance ratio
Ecoli	336	7C	imU	1:9
Spectrometer	531	93C	LRS >=44	1:11
Balance	625	4N	Balance	1:12
Libras Move	360	90C	Positive	1:14
Arrhythmia	452	73N, 206C	Class=06	1:17
Car Eval.	1728	6N	Very good	1:25
Yeast	1484	8C	ME2	1:28
Abalone	4177	1N, 7C	Ring=19	1:130

All algorithms are implemented in the MATLAB framework. In all experiments, 5-fold stratified cross validation is applied. 5-fold cross validation is chosen due to limited number of minority instances in most datasets. The whole process of cross validation is repeated ten times and the final outputs are the means of these ten runs.

Decision trees have been commonly used in several imbalanced problems as a base classifier [5], [11], [12]. In this paper, the CART algorithm [10] is chosen as base learning method for our experiments.

We applied the K-means clustering algorithm to partition majority instances. However, instead of using the Euclidean distance to find similarity of instances, the L1-norm is used. The advantage of using the L1-norm over the Euclidean distance is that it is less sensitive to outliers in the data. Also, the probability of having a singleton partition for outliers is less than Euclidean distance [14]. Further, it has been shown that using the L1- norm is more suitable when learning in a setting which is susceptible to class imbalance, especially where the number of features is higher that the number of minority class examples [18].

4.3 Results and Analyses

In the first experiment, we evaluate the performance of our proposed single classifiers. Table 3 demonstrates the results of applying these methods on our real datasets and comparing them in terms of F-measure and G-mean. The results show that NearMiss1 has better performance compared to other classifiers and the classifier that uses the centroids as cluster representative has significantly lower F-measure and G-means. It can be concluded that cluster centroids are not informative for our classification task. In summary, as we expected, single undersampling learner suffers from information loss. In the rest of the experiments, we use an ensemble-learning method instead of using a single CART classifier.

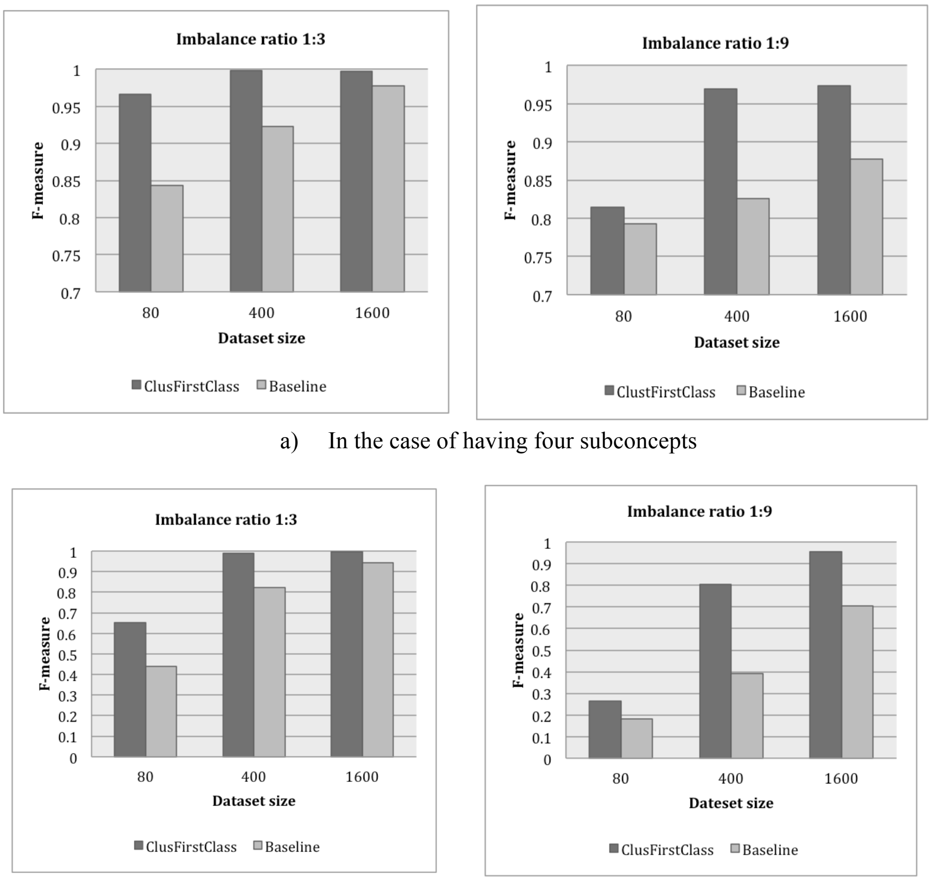
In the next experiment, to evaluate our proposed ensemble classifier ClusFirstClass in different scenarios with different degree of within and between class imbalances, it is applied on artificial datasets. We consider a bagging ensemble learner that chooses randomly a subset of majority instances to be combined with all minority instances, as the baseline and compare the performance of this algorithm with our proposed method. The only difference between baseline method and ClusFirstClass is that it chooses majority instances randomly. It has the same number of base learners and combination rule.

Table 3. F-measure and G-mean of proposed single classifiers

Dataset	**F-Measure**				**G-Mean**			
	Near Miss1	Near Miss2	Most-Distant	Centroid	Near Miss1	Near Miss2	Most-Distant	Centroid
Ecoli	0.5915	0.5671	0.5392	0.5740	0.8433	0.8261	0.8357	0.8527
Spectrometer	0.4874	0.5180	0.4746	0.4709	0.8488	0.8445	0.8488	0.8488
Balance	0.1448	0.1417	0.1802	0.1415	0.5050	0.4600	0.5551	0.0581
Libras Move	0.3945	0.3433	0.3154	0.3789	0.8147	0.7865	0.7708	0.7991
Arrhythmia	0.3594	0.3074	0.3342	0.3646	0.7855	0.7411	0.7623	0.7737
Car Eval.	0.8313	0.8057	0.2778	0.2394	0.9730	0.9857	0.8907	0.8513
Yeast	0.2398	0.1999	0.1928	0.1138	0.7827	0.7657	0.7872	0.5412
Abalone	0.0301	0.0328	0.0275	0.0174	0.6496	0.6820	0.6599	0.2873
Average	**0.3849**	0.3645	0.2927	0.2876	**0.7753**	0.7614	0.7638	0.6265

Figure 2 and 3 illustrates the results of applying our proposed method on previously described uni-dimensional and multi-dimensional artificial datasets respectively. In all 12 scenarios ClusFirstClass is compared to baseline method in terms of F-measure. For all datasets, our proposed method has considerably better performance compared to baseline. In most cases, as the imbalance ratio and the number of subconcepts increase the difference between our proposed classifier and baseline algorithm becomes more significant.

In the first experiment with proposed single classifiers, we have clustered majority instances into k groups using the k-means algorithm, where *k*=|*Dmin*|. In experiments on artificial datasets, obviously the number of clusters is equal to the number of subconcepts within the majority class. For the next experiments, in order to compute the natural number of clusters, different number of k from 1 to |*Dmin*| is tested to find the one with the best average Silhouette plot [17].

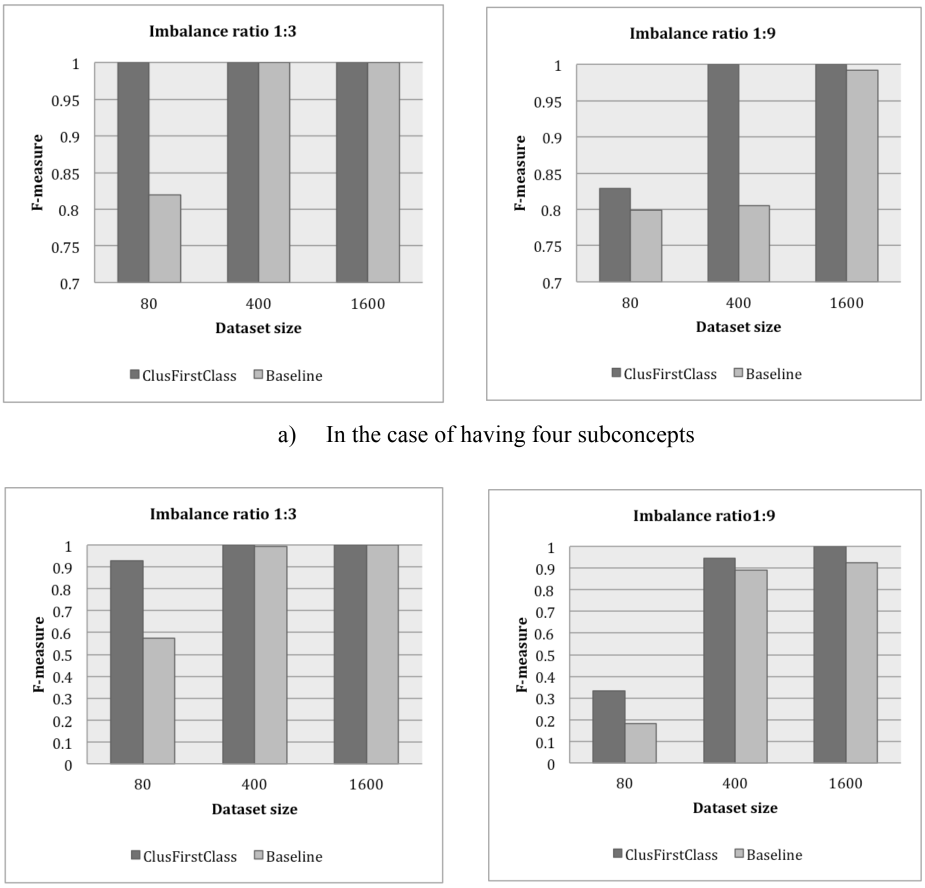


a) In the case of having four subconcepts

b) In the case of having eight subconcepts

Fig. 2. Results of applying our proposed method on previously described uni-dimensional artificial datasets in terms of F-measure

Table 4 shows the results of comparing our proposed undersampling ensemble algorithm based on clustering with another undersampling ensemble method, EasyEnsemble [5] and two cluster-based algorithms, Cluster-based oversampling [2] and SBC [6]. Our algorithm outperforms other undersampling ensemble methods on almost all datasets in terms of F-Measure and G-Mean. Compared to the cluster-based oversampling, although that method achieves better F-measure on two (out of 9) datasets, the averaged F-measure and G-Mean of our algorithm is better than that of Cluster-based oversampling. Clust-First-Class outperforms SBC on all datasets.



a) In the case of having four subconcepts

b) In the case of having eight subconcepts

Fig. 3. Results of applying our proposed method on previously described multi-dimensional artificial datasets in terms of F-measure

5 Conclusion and Future Work

In this paper, we introduce a new cluster-based classification framework for learning from imbalanced data. In our proposed framework, first majority instances are clustered into k groups and then at least one instance from each cluster is selected to combine with all minority instances, prior to training. This approach is capable of handling between-class imbalance by selecting approximately the same number of instances from minority and majority classes. Further, we address within-class imbalance by focusing on all clusters equally. Finally, to reduce information loss due to choosing small number of majority instances in highly imbalanced datasets, we employ an ensemble learning approach to train several base learners with different subsets of majority instances. An advantage of our ClustFirstClass method is that we guide the selection of majority instances used during training, as based on the clusters obtained by the k-means algorithm

To evaluate the efficiency of our proposed method, it is applied on two sets of artificial datasets with varying degree of between class imbalances and different number of subconcepts. For all datasets, our proposed method has considerably better performance compared to the baseline method. In most cases, as the imbalance ratio and the number of subconcepts increase, the difference between our proposed classifier and baseline algorithm becomes more significant. Experimental results on real datasets demonstrate that our proposed ensemble learner has better performance than our proposed single classifiers. It also shows that our suggested ensemble method yields promising results compared to other state-of-the-art methods in terms of G-means and F-measure.

Several directions of future research are open. Our experimental results indicate that using the K-means algorithm yield encouraging results. However, we are interested in exploring other cluster analysis algorithms, since the K-means algorithm may not be ideal when considering highly imbalanced datasets [19], or when considering extending our work to the multi-class problems. Thus, we plan to investigate the use of more sophisticated clustering algorithms to partition the majority instances. Another direction would be to consider other ensemble-based techniques. In particular, ECOC [15] may be a favorable choice as it targets performance improvement in a binary classification setting. We also plan to extend our experiments with more datasets and compare it with more ensemble algorithms such as RB bagging [20] and also testing other base learning algorithms such as SVM and KNN.

Table 4. F-measure and G-mean of proposed ensemble classifier, ClustFirstClass, compared to EasyEnsemble, Cluster-oversampling and SBC methods

Dataset	**F-Measure**				**G-Mean**			
	Clust First Class	Easy Ensemble	Clust Over Sample	SBC	Clust First Class	Easy Ensemble	Clust Over Sample	SBC
Ecoli	**0.5961**	0.5612	0.5088	0.5140	**0.8689**	0.8658	0.6899	0.8489
Spectrometer	0.5944	**0.6924**	0.6740	0.4485	0.8878	**0.9064**	0.8053	0.8186
Balance	**0.1524**	0.0793	0.0290	0.1508	**0.5223**	0.3452	0.0967	0.4971
Libras Move	0.5912	0.4806	**0.6652**	0.4258	**0.8451**	0.8407	0.8006	0.7372
Arrhythmia	**0.7475**	0.6360	0.5757	0.5996	**0.9489**	0.8802	0.7548	0.9219
Car Eval.	**0.8331**	0.3613	0.9566	0.6892	**0.9918**	0.9237	0.9812	0.9792
Yeast	0.2720	0.2613	**0.2798**	0.2065	**0.8054**	0.8044	0.5095	0.8016
Abalone	**0.0449**	0.0381	0.0618	0.0315	**0.7446**	0.7309	0.1903	0.6794
Average	**0.4790**	0.3888	0.4689	0.3832	**0.8267**	0.7872	0.6035	0.7855

References

1. Japkowicz N., Stephen S.: The class imbalance problem: A systematic study. J. Intelligent Data Analysis 6(5), 429–450 (2002)
2. Jo T., Japkowicz N.: The class imbalance versus small disjuncts. ACM SIGKDD Exploration Newsletter 6(1), 40-49 (2004)
3. Sun Y., Kamel M.S., Wong A.K.C., Wang Y.: Cost-Sensitive Boosting for Classification of Imbalanced Data. J. Pattern Recognition 40(12), 3358-3378 (2007)
4. He H., Garcia E.: Learning from imbalanced data. J. IEEE Transactions on Data and Knowledge Engineering 9(21), 1263–1284 (2009)
5. Liu X.Y., Wu J., Zhou Z.H.: Exploratory Under Sampling for Class Imbalance Learning. Proc. Int'l Conf. Data Mining, pp. 965-969, (2006)
6. Yen, Lee: Cluster-based under-sampling approaches for imbalanced data distributions, Expert Systems with Applications: An International Journal Volume 36 Issue 3, April, (2009)
7. Zhang J., Mani I.: KNN Approach to Unbalanced Data Distributions: A Case Study Involving Information Extraction. Proc. Int'l Conf. Machine Learning (ICML '2003), Workshop Learning from Imbalanced Data Sets (2003)
8. Ding, Zejin: Diversified ensemble classifiers for highly imbalanced data learning and its application in bioinformatics. Phd thesis, Georgia State University (2011)
9. Bache K., Lichman M.: UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science (2013)
10. Breiman L., Friedman J., Olshen R., Stone C.: Classification and Regression Trees. Boca Raton, FL: CRC Press (1984)
11. Batista G., Prati RC, Monard MC.: A study of the behaviour of several methods for balancing machine learning training data. SIGKDD Explor 6(1), 20–29 (2004)
12. Bianchi N. et al.: Synergy of multi-label hierarchical ensembles, data fusion, and cost-sensitive methods for gene functional inference, Machine Learning 88(1), 209-241 (2012)
13. Błaszczyński J. et al.: Extending Bagging for Imbalanced Data. In *Proceedings of the 8th International Conference on Computer Recognition Systems,* pp. 269-278 (2013)
14. Manning C., Schutze H.: Foundations of Statistical Natural Language Processing. MIT Press Cambridge, MA (1999)
15. Dietterich T. G., Bakiri G.: Solving Multiclass Learning Problems via Error-Correcting Output Codes. J. AI Research 2, 263-286 (1995)
16. Galar M. et al: A Review on Ensembles for the Class Imbalance Problem: Bagging-, Boosting-, and Hybrid-Based Approaches Systems. Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions, 42(4) (2012)
17. Rousseeuw Peter J.:Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis. Computational and Applied Mathematics 20, 53–65 (1987)
18. Ng A.: Feature selection, L1 vs. L2 regularization and rotational invariance. 21st International Conference on Machine Learning (2004)
19. Coates A., Ng A.: Learning Feature Representations with K-means, In Neural Networks: Tricks of the Trade (Second Edition), Springer LNCS 7700, 561-580 (2012)
20. Shohei H., Hisashi K., Yutaka T.: Roughly balanced bagging for imbalanced data, Statistical Analysis and Data Mining, 2, 5-6, 412-419 (2009)