Efficient Resampling Methods for Training Support Vector Machines with Imbalanced Datasets

Rukshan Batuwita, Vasile Palade

***Abstract*—Random undersampling and oversampling are simple but well-known resampling methods applied to solve the problem of class imbalance. In this paper we show that the random oversampling method can produce better classification results than the random undersampling method, since the oversampling can increase the minority class recognition rate by sacrificing less amount of majority class recognition rate than the undersampling method. However, the random oversampling method would increase the computational cost associated with the SVM training largely due to the addition of new training examples. In this paper we present an investigation carried out to develop efficient resampling methods that can produce comparable classification results to the random oversampling results, but with the use of less amount of data. The main idea of the proposed methods is to first select the most informative data examples located closer to the class boundary region by using the separating hyperplane found by training an SVM model on the original imbalanced dataset, and then use only those examples in resampling. We demonstrate that it would be possible to obtain comparable classification results to the random oversampling results through two sets of efficient resampling methods which use 50% less amount of data and 75% less amount of data, respectively, compared to the sizes of the datasets generated by the random oversampling method.**

I. Introduction

Class imbalance problem is commonly found in various machine learning applications [1]-[3]. In this paper we consider binary classification problem, where the positive class is treated as the minority class while the negative class is treated as the majority class. Support Vector Machines (SVMs) is a very popular machine learning algorithm due to its solid theoretical background, ability to find global classification solutions and high generalization capabilities [4]. Although SVMs work effectively with balanced datasets, they provide sub-optimal models with imbalanced datasets [5][6]. It has been identified that when an SVM model is developed with an imbalanced dataset, often, the separating hyperplane found can be skewed towards the minority class, which can result in a large number of false negative predictions [6]. This effect would lead to the development of models having low positive recognition rates (SE=Sensitivity) and high negative recognition rates (SP=Specificity).

Among the available class imbalance learning techniques, random undersampling and oversampling are simple yet very popular resampling methods [1]-[3]. In random undersampling, the examples from the majority class are removed until the datasets are balanced. In random oversampling, the minority class examples are randomly duplicated to balance the datasets. The random undersampling method has been criticized in some papers stating that the random removal of a large percentage of negative examples can result in a huge information loss [1][7][8]. The oversampling method does not possess any information loss, and hence, could produce better results. However, due to the addition of new training examples, the oversampling increases the computational requirements needed to develop machine learning models largely. This increase in computational cost would be significant for SVMs, since the standard SVM learning has $O(l^3)$ time complexity, where l is the number of training examples [9][10].

The main purpose of this paper is to propose an efficient resampling method from which we would be able to obtain comparable classification results to the random oversampling results, but with the generation of less amount of data. The main idea of the proposed method is to first select the most informative negative examples, and then apply random oversampling to duplicate positives to match with the number of selected negatives, rather than blindly oversampling positives to match with all the negatives as in the random oversampling method. In the initial experiments we considered that the examples located around the class boundary are the most informative ones, and used the separating hyperplane found by SVM learning on the original imbalanced dataset to select the most informative examples. During these experiments we identified that what matters the most in class imbalance learning is not the difference in the number of positive and negative examples, but the difference in the distributions of them (i.e. the positive and negative data densities) around the class boundary. Based on these observations, we introduced several efficient resampling methods from which we were able to obtain comparable classification results to the random oversampling results, first by using 50% less amount of data and then by using 75% less amount of data compared to the size of the datasets generated by the random oversampling method. We evaluated all the experiments carried out in this

R. Batuwita is with the Oxford University Computing Laboratory, Oxford University, Wolfson Building, Parks Road, Oxford, OX1 3QD, UK. (e-mail: manb@comlab.ox.ac.uk)

V. Palade is with the Oxford University Computing Laboratory, Oxford University, Wolfson Building, Parks Road, Oxford, OX1 3QD, UK. (e-mail: vasile.palade@comlab.ox.ac.uk).

978-1-4244-8126-2/10/$26.00 ©2010 IEEE

research on five real-world imbalanced datasets.

This paper is organized as follows. First, in section 2 we introduce the imbalanced datasets used in the experiments conducted in this research. Then in section 3 we explain the problems associated with the conventional random undersampling and oversampling methods through the experimental results obtained on the datasets considered. Section 4 introduces the main idea of the proposed efficient resampling methods. Then Section 5 details the proposed resampling methods to reduce the oversampling data by 50% with the experimental results obtained. Afterwards, Section 6 discusses the proposed resampling methods to reduce the data by 75% and the results obtained. Finally, the conclusions are drawn in Section 7.

II. Imbalanced Datasets Considered

For the experiments carried out in this research we considered four large imbalanced Bioinformatics datasets (*miRNA, Splice-site, Promoter* and *Drug-like*) used in our previous research in class imbalance learning [11]. In addition, we considered the *Pageblocks* dataset from the UCI machine learning repository [12]. These five datasets are described below:

miRNA dataset: this dataset was used in our previous research [13][14], which is available at http://web.comlab.ox.ac.uk/people/ManoharaRukshan.Batuwita/microPred.htm. This dataset contains 691 positives (real miRNA hairpins) and 9248 negatives (negative hairpins), which was represented by 21 features.

Splice-site dataset: This dataset originates from http://www.fruitfly.org/sequence/human-datasets.html. We considered the training and testing data splits as given in the original dataset, where the training dataset contains 1116 positives and 4672 negatives, while the testing dataset contains 208 positives and 881 negatives. This dataset was represented by the statistical features introduced in [15].

Promoter dataset: This human promoter dataset is available at http://www.fruitfly.org/sequence/human-datasets.html, which contains 565 promoters, 890 cds and 4345 introns. After removing the sequences containing missing bases, we recovered 471 promoters as the positive dataset, and combined 840 cds and 4241 introns as the negative dataset. This dataset was represented by 16 features [11].

Drug-target dataset: This dataset has been constructed and used in [16], which is composed of 521 drug target proteins (positives) and 5019 non-target proteins (negatives). This dataset was represented by 290 features.

Page-blocks dataset: This dataset was obtained from the UCI machine learning repository. Out of the 5 classes available in this dataset, the examples belonged to the class number 5 were selected as the positive dataset, while the examples belonged to the remaining classes were combined as the negative dataset. This resulted in 115 positives and 5358 negatives represented by 10 features.

III. Problems of the Existing Resampling Methods

As mentioned earlier, the random undersampling method has been criticized in the literature stating that it can remove a lot of informative examples which could be useful in the development of the classifiers [1][7][8]. In this section we investigate the problems associated with random undersampling and oversampling methods through the classification results obtained on the above datasets.

For each dataset, we trained SVM models, first, with the original imbalanced dataset, and then with the datasets generated by the random undersampling and oversampling methods. We used the *libsvm* package [17] as the SVM learning environment. For the *miRNA, Promoter, Drug-like* and *Pageblocks* datasets we evaluated the performance of SVM models through an extensive five fold cross-validation method. For the splice-site dataset, we used the training and testing datasets as they were given in the original dataset. As the performance measure, we considered the Geometric mean of SE and SP ($Gm = \sqrt{SE \cdot SP}$) as commonly used in class imbalance learning research [5][6][13].

In these experiments, the random oversampling and undersampling methods were applied to the training dataset partitions until they were balanced and left the testing dataset partitions in their original imbalanced distributions. Due to the randomness associated with these resampling methods, each method was repeated five times in each cross-validation run and the results obtained on the testing partitions were averaged. The classification results obtained from these initial experiments are given in Table 1. The columns dSE and dSP represent the amount of SE increased and the amount of SP decreased by applying the corresponding resampling method compared to the normal SVM results. Table 1 also compares the sizes of the training datasets produced by the resampling methods with the size of the original imbalanced datasets.

From these results we could first see that all the imbalanced datasets resulted in sub-optimal models having high SP and low SE in normal SVM training. Then when the undersampling and oversampling methods were applied, SE was increased and SP was decreased in different amounts for different datasets (generally, this happens when applying class imbalance learning methods as increasing both SE and SP simultaneously are two contradictory goals). For all the datasets considered we could observe that the undersampling method decreased SP more than the oversampling method did. On the other hand, we also observed that the random undersampling increased SE more than the random oversampling method did. Therefore, from these raw results we would not be able to directly see any effect of the 'information loss' caused by the random undersampling method. Also, based on the Gm values we could not observe any significant difference between random oversampling and undersampling results.

TABLE 1.COMPARISON OF THE CLASSIFICATION RESULTS OBTAINED BY THE RANDOM UNDERSAMPLING AND OVERSAMPLING METHODS WITH THE NORMAL SVM RESULTS.

Dataset	Method	Classification results						Size of training dataset		
		SE	SP	Gm	dSE	dSP	R	Pos.	Neg.	Total
miRNA	Normal	82.78	99.45	90.73				533	7398	7931
	Undersampling	91.03	93.02	92.02	+8.25	-6.43	0.78	533	533	1066
	Oversampling	89.93	96.53	93.17	+7.15	-2.92	0.41	7398	7398	14796
Promoter	Normal	25.69	98.75	50.37				377	4105	4482
	Undersampling	70.08	80.67	75.19	+44.39	-18.78	0.42	377	377	754
	Oversampling	68.89	82.56	75.42	+43.20	-16.89	0.39	4105	4105	8210
Splice-site	Normal	73.56	96.71	84.34				1116	4672	5788
	Undersampling	89.33	89.74	89.53	+15.77	-6.97	0.44	1116	1116	2232
	Oversampling	87.88	91.12	89.49	+14.32	-5.59	0.39	4672	4672	9344
Drug-like	Normal	71.59	97.15	83.40				417	4015	4432
	Undersampling	90.75	84.67	87.66	+19.26	-12.48	0.65	417	417	834
	Oversampling	88.29	88.94	88.61	+16.70	-8.21	0.49	4015	4015	8030
Pageblocks	Normal	58.26	99.52	76.14				92	4286	4378
	Undersampling	93.74	92.52	93.13	+35.48	-6.99	0.20	92	92	184
	Oversampling	91.83	94.88	93.34	+33.57	-4.63	0.14	4286	4286	8572

Therefore, in order to compare the results produced by these resampling methods with each other, we introduced a simple measure: $R = |dSP / dSE|$. R represents the amount of SP reduced when the SE is increased by 1%, through applying a particular resampling method for a particular dataset. Based on the R values given in Table 1, we could see that for each dataset random oversampling method resulted in a lower R value than the random undersampling method. That is, the random undersampling method sacrificed a higher amount of SP than the random oversampling method in order to increase the SE by 1%. We believe that this high reduction of SP in undersampling occurred due to the removal of a large percentage of negative examples randomly, which could discard a lot of informative negatives (i.e. the information loss).

Generally, for any imbalanced classification problem, by applying a class imbalance learning method it is vital to increase SE as much as possible with a less amount of reduction of SP in order to obtain a high overall classification result (i.e., a high SE and a high SP). Based on the above results, we can argue that the oversampling method could result in better overall classification results than the undersampling method, since the oversampling could increase SE with a lower rate of reduction of SP than the undersampling method. However, as we can observe, the major problem generated by the oversampling method is the large increase of training dataset size due to the addition of new positive examples to balance the datasets (compare the sizes of the datasets generated by the oversampling method to the original imbalanced dataset given in Table 1). This would significantly increase the computational power required to train SVM models whose standard time complexity is $O(l^3)$.

IV. PROPOSED METHOD: THE MAIN IDEA

As we showed in the previous section, although the oversampling method could produce better classification results than the undersampling method, it increases the required computational power to train a classification model hugely due to the addition of new minority class examples to balance the datasets. In this study we mainly investigated to develop an efficient resampling method from which we would be able to obtain comparable classification results to the oversampling results, but with reduced computational cost. The main idea of the proposed method was to first select only the most informative negative examples and then apply oversampling to balance them, rather than blindly oversampling the positive examples to match with the total number of negative examples. As the initial idea, we treated the examples located closer to the class boundary as the most informative ones as these are the examples contributing mostly when finding the separating hyperplane in SVM learning. That is, in SVM learning, usually the examples located closer to the class boundary are selected as the support vectors. Therefore, we used the separating hyperplane found by applying the SVM algorithm on the original imbalanced dataset to identify the examples located closer to the class boundary. Hereafter, we refer to this hyperplane as the *imbalanced hyperplane*. We assumed that this *imbalanced hyperplane* is still located around the actual class boundary, although it can be skewed towards the minority class due to the effect of class imbalance, as mentioned previously.

V. REDUCTION OF DATA BY 50%

First, we considered a resampling method that generated a dataset which was half the size of the dataset generated by

the random oversampling method. In these experiments we first selected 50% of negative examples in different ways and then oversampled the positives to match with the selected negatives. With 50% reduction of data, we would be able to reduce the theoretical upper-bound of the SVM time complexity to $O(l^3/8)$.

A. The closest 50% negatives

In this initial experiment we first selected the 50% of negative examples located closest to the *imbalanced hyperplane* and all the positive examples. Then we randomly over-sampled the positives to match with the selected number of negatives. We named this method as '*closest-50%-over'* method. We applied this method to balance the datasets considered, developed SVM classifiers and evaluated their results. These results are given in Table 2. Here we directly compare the SE and SP obtained by the proposed method with the SE and SP obtained by the random oversampling method.

TABLE 2. COMPARISON OF THE CLASSIFICATION RESULTS OBTAINED BY THE *CLOSEST-50%-OVER* METHOD WITH THE RANDOM OVERSAMPLING RESULTS.

Dataset	Method	Results (%)	
		SE	SP
miRNA	Normal	82.78	99.45
	Oversampling	89.93	96.53
	Closest-50%-over	87.39	98.32
Promoter	Normal	25.69	98.75
	Oversampling	68.89	82.56
	Closest-50%-over	62.64	86.96
Splice-site	Normal	73.56	96.71
	Oversampling	87.88	91.12
	Closest-50%-over	82.12	94.44
Drug-like	Normal	71.59	97.15
	Oversampling	88.29	88.94
	Closest-50%-over	80.99	93.68
Pageblocks	Normal	58.26	99.52
	Oversampling	91.83	94.88
	Closest-50%-over	86.09	95.97

From the classification results presented in Table 2 we can see that the proposed method did not manage to yield comparable classification results with the random oversampling results. That is, for all the datasets the *closest-50%-over* method did not increase SE as much as the random oversampling method did. In order to find out the reason behind this problem, we closely investigated the distributions of the dataset generated by the *closest-50%-over* method with respect to the *imbalanced-hyperplane*.

The distributions of the original imbalanced dataset, the dataset generated by the random oversampling method and the dataset generated by the *closest-50%-over* method for the *miRNA* dataset are depicted in Figure 1. These distributions for the *Splice-site* dataset are given in Figure 2. These figures show the frequencies of training examples located in different distances from the *imbalanced-hyperplane*. Since the geometric distance of a training example is directly proportional to its SVM decision value [4], actually what we plotted in these figures are the frequencies of training examples (in y axis) against their decision values with respect to the *imbalanced-hyperplane* (in x axis). The distribution of the positive dataset is represented by a solid line while the distribution of the negative dataset is represented by a dotted line.

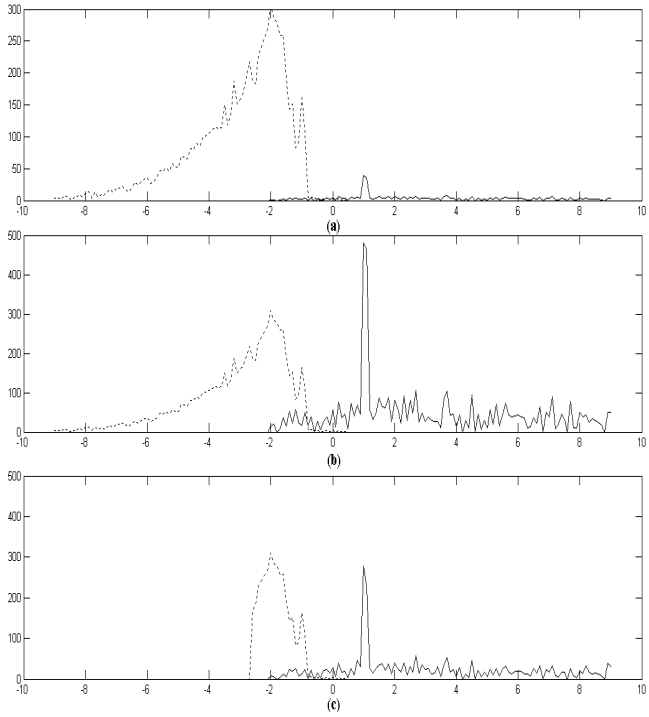

Figure 1. Distributions of the miRNA dataset (a). Original imbalanced dataset. (b). Dataset generated by the random oversampling method. (c). Dataset generated by the *closest-50%-over* method.

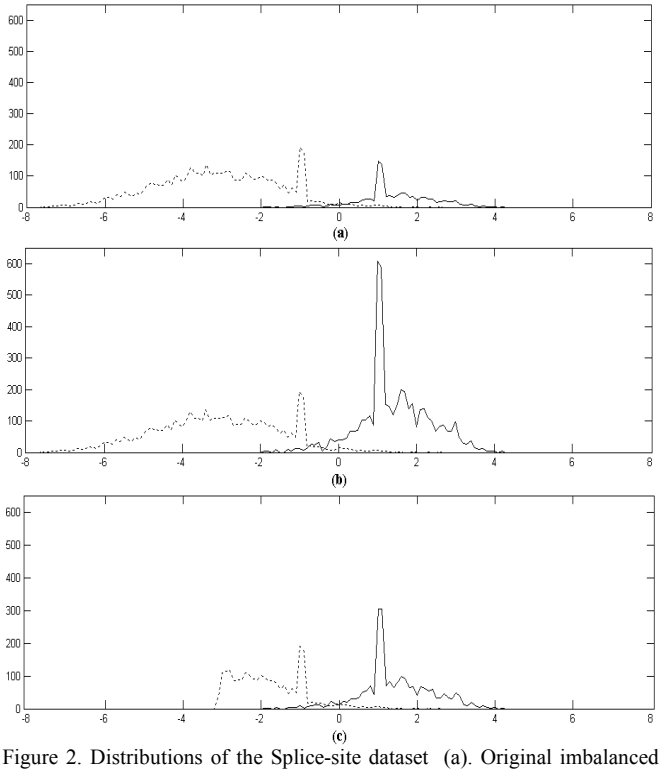

Figure 2. Distributions of the Splice-site dataset (a). Original imbalanced dataset. (b). Dataset generated by the random oversampling method. (c). Dataset generated by the *closest-50%-over* method.

From Figure 1.(a) and Figure 2.(a) for each original imbalanced dataset we can clearly see the higher negative density compared to the positive density, which caused the development of sub-optimal models having low SE. Then we can see that when the oversampling method was applied to balance a dataset the positive density was increased largely (Figure 1.(b) and Figure 2.(b)), which caused the development of models having increased SE but with some reduction of SP.

Next we compare the distribution of the dataset generated by the *50%-closest-over* method with the distribution of the dataset generated by the random oversampling method. From this comparison we can observe that the *50%-closest-over* method did not manage to increase the positive density around the separating hyperplane as much as the random oversampling method did. However, the 50% of selected negatives closest to the separating hyperplane by this method still possessed the same high original negative density around the separating hyperplane (and hence, around the class boundary region) as the negative density of the dataset selected by the oversampling method. Due to this reason *50%-closest-over* method was unable to increase the SE as much as random oversampling method did. These observations indicate the importance of the distribution of negative and positive data densities around the class boundary in class imbalance problem. In order to investigate this matter further we conducted the experiment explained in the following section.

B. Investigating the effect of class densities around the class boundary region

In this section we explain an experiment carried out to further investigate the effect of the distributions of positive and negative densities around the class boundary in class imbalance problem. Here we applied a focused undersampling method which selected the negative examples located closest to the *imbalanced hyperplane* to match with the number of positive examples in the dataset. For example, for the *miRNA* dataset we selected 533 negative examples located closest to the *imbalanced-hyperplane* to balance the 533 positive examples in the training dataset. We named this method as '*closest-under'* method. We applied this method to balance the datasets considered, and then developed SVM classifiers and evaluated their performance. The classification results obtained by this focused undersampling method are compared with the original imbalanced results and the random undersampling results given in Table 3.

From these results we can observe that the proposed *closest-under* method did not manage to increase the SE any closer to the amount of SE increased by the random undersampling method. In other words, the results given by the *closest-under* method are more comparable to the original imbalanced classification results, which were obtained by the normal SVM training, than to the random undersampling results. In fact, for the *miRNA, Splice-site* and *Drug-like* datasets, results given by the *closest-under* method are more similar to the results given by the original imbalanced classification results. In order to investigate the cause of this problem we closely compared the distributions of the datasets generated by the *closest-under* method with the distribution of the dataset generated by the random undersampling method and the original imbalanced dataset for the *miRNA* dataset. These distributions are depicted in Figure 3.

TABLE 3. COMPARISON OF THE CLASSIFICATION RESULTS OBTAINED BY THE *CLOSEST-UNDER* METHOD WITH THE RANDOM UNDERSAMPLING AND NORMAL SVM RESULTS.

Dataset	Method	Results (%)	
		SE	SP
miRNA	Normal	82.78	99.45
	Undersampling	91.03	93.02
	Closest-under	81.34	99.36
Promoter	Normal	25.69	98.75
	Undersampling	70.08	80.67
	Closest-under	43.53	93.47
Splice-site	Normal	73.56	96.71
	Undersampling	89.33	89.74
	Closest-under	73.56	96.71
Drug-like	Normal	71.59	97.15
	Undersampling	90.75	84.67
	Closest-under	72.93	96.94
Pageblocks	Normal	58.26	99.52
	Undersampling	93.74	92.52
	Closest-under	69.57	97.83

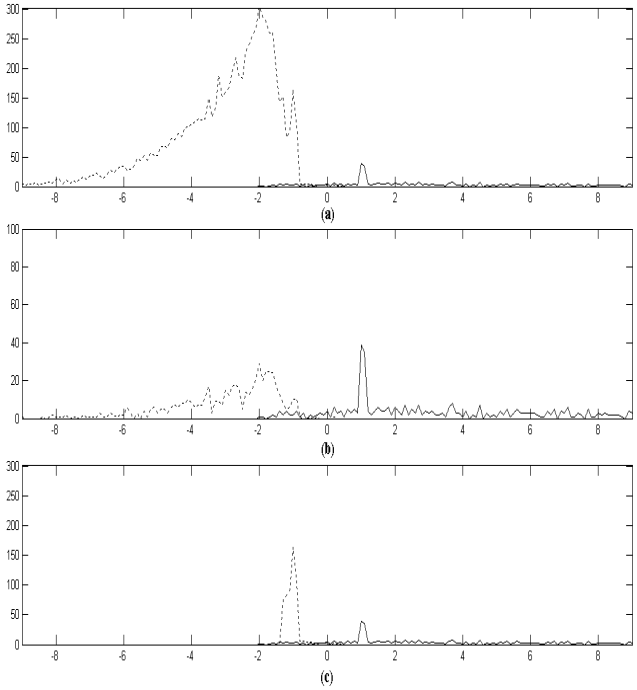

Figure 3. Distributions of the miRNA dataset (a). Original imbalanced dataset. (b). Dataset generated by the random undersampling method. (c). Dataset generated by the *closest-under* method.

From Figure 3.(b) we can see that random undersampling

method decreased the negative density around the hyperplane largely compared to the original imbalanced distribution given in Figure 3.(a). Therefore, the dataset generated by random undersampling method resulted in models having high increase of SE, but at the cost of large decrease of SP. When we observe the distribution of the dataset generated by the *closest-under* method given in Figure 3.(c), we can see that the selected closest negatives to the hyperplane (to match with the number of positive examples) still possessed a large negative density very close to the separating hyperplane (and hence around the class boundary region) like in the original imbalanced dataset. We found this same effect when we observed the distributions of the other datasets considered in this study. Due to this high negative density around the class boundary, despite the fact that the positive and negative datasets were balanced, *closest-under* method was unable to increase the SE much or at all, and produced comparable results to the original imbalanced learning results.

The overall findings in this experiment further convinced us that *what matters the most in class imbalance learning is not the imbalance in the number of positive and negative training examples, but the imbalance in the distribution of their densities with respect to the class boundary*. Related to this we further identified that selecting data located closer to the class boundary by treating that they are more informative alone would not solve the class imbalance problem.

C. Improving the proposed resampling method

As we observed earlier, the proposed efficient resampling method, *50%-closest-over*, did not manage to produce comparable results with the oversampling results due to the higher negative density of the generated dataset around the class boundary. Therefore, in order to obtain comparable classification results with the oversampling results we looked into two solutions of improving the proposed resampling method. One method was to select 50% of negative examples located closest to the separating hyperplane as previously and increase the positive density closer the hyperplane by oversampling. As we closely observed the distributions of the original imbalanced datasets for all these five datasets we could see that they were distributed in different ranges with respect to the corresponding separating hyperplane. For examples, the positive examples in the *miRNA* dataset (Figure 1.(a)) are distributed in a wider range than the distribution of the positive examples in the *Splice-site* dataset (in Figure 2.(a)). Therefore, it would not be possible to find a common range around the hyperplane, which would be effectively suitable for all the datasets, to over-sample the positives in order to obtain a higher positive density. Therefore, we did not consider this method as a solution to our problem.

The other solution was to select 50% of negative examples having less negative density around the hyperplane than the density of the closest 50% negatives, and over-sample the positives in the normal way to balance the dataset. Since this method could be applied to all the datasets considered without any problem, we considered this method in our study. Based on this idea we proposed following two resampling methods:

1) 75%-under(0.33)-over

In this method we selected 75% of negatives located closest to the *imbalanced-hyperplane*, and removed 1/3 of them by random undersampling to retain 50% of the total negatives. From this way we could select 50% of negatives having less density around the hyperplane than the 50% of closest negatives. Then we randomly over-sampled all the positives to balance the dataset.

2) 100%-under(0.5)-over

In this method we first selected all the negative examples in the dataset and removed half of them by random undersampling. Then we considered all the positives and applied random oversampling to balance the dataset.

We applied these two methods to balance all the datasets considered. Then we developed SVM classifiers and evaluated their results through 5-fold cross-validation learning. Due to the randomness involved, each of this proposed resampling method was repeated five times for each cross-validation run and the results were averaged. The results obtained in these experiments are compared with random oversampling results in Table 4.

TABLE 4. COMPARISON OF THE CLASSIFICATION RESULTS OBTAINED BY THE PROPOSED RESAMPLING METHODS WHICH REDUCE DATA BY 50%

Dataset	Method	Results		
		SE	SP	ED
miRNA	Oversampling	**89.93**	**96.53**	
	Undersampling	91.03	93.02	3.68
	Closest-50%-over	87.39	98.32	3.11
	75%-under(0.33)-over	89.00	97.23	1.16
	100%-under(0.5)-over	**90.74**	**96.16**	**0.89**
Promoter	Oversampling	**68.89**	**82.56**	
	Undersampling	70.08	80.67	2.23
	Closest-50%-over	62.64	86.96	7.64
	75%-under(0.33)-over	66.05	85.36	3.99
	100%-under(0.5)-over	**69.23**	**81.13**	**1.46**
Splice-site	Oversampling	**87.88**	**91.12**	
	Undersampling	89.33	89.74	2.00
	Closest-50%-over	82.12	94.44	6.65
	75%-under(0.33)-over	86.25	92.74	2.29
	100%-under(0.5)-over	**88.75**	**90.83**	**0.92**
Drug-like	Oversampling	**88.29**	**88.94**	
	Undersampling	90.75	84.67	4.93
	Closest-50%-over	80.99	93.68	8.70
	75%-under(0.33)-over	87.14	89.82	1.44
	100%-under(0.5)-over	**88.29**	**88.44**	**0.50**
Pageblocks	Oversampling	**91.83**	**94.88**	
	Undersampling	93.74	92.52	3.04
	Closest-50%-over	86.09	95.97	5.84
	75%-under(0.33)-over	87.83	95.17	4.01
	100%-under(0.5)-over	**89.57**	**94.90**	**2.26**

In order to compare the SE and SP obtained from a proposed resampling method (SE_i, SP_i) with the SE and SP given by the random oversampling method (SE_o, SP_o), we used the Euclidean distance (ED) between the results $(ED = \{(SE_o - SE_i)^2 + (SP_o - SP_i)^2\}^{1/2})$. Here we did not consider any unit performance measure, such as Gm, to compare the results, since the same value in Gm can be resulted by different combinations of the SE and SP values. For example, (SE=92%, SP=98%) and (SE=98%, SP=92%) would result in the same Gm value which is 94.95%. In Table 4 we also present the results obtained by the random undersampling method and the *50%-closest-over* method for the comparative reasons. The random oversampling results and the closest results to them are depicted in bold type.

Among all the resampling methods considered so far, from the *100%-under(0.5)-over* method we were able to obtain the most comparable classification results to the oversampling results by using only half of the amount of data used by the random oversampling method.

VI. Reduction of Data by 75%

We further extended our experiments to observe whether we could obtain comparable classification results to the oversampling results by reducing the amount of data by 75%. The main idea of this method would be to select 25% of negative example and oversample the positives to balance the dataset. With 75% reduction of data the SVM training complexity could be further reduced to $O(N^3/64)$.

Based on the results obtained from the *50%-closest-over* and *closest-under* method in the previous experiments, we did not consider the *closest-25%-over* method (i.e. selecting the 25% negatives closest to the *imbalanced hyperplane* and oversampling the positives to balance the dataset) as it would not give comparable classification results to the oversampling results due to the high negative density of the selected examples. In contrast, we considered the following resampling methods which deal with the selection of 25% of negative examples having less negative density around the *imbalanced hyperplane* in different ways and oversample the positives to balance the datasets.

1) 50%-under(0.5)-over

In this method we first selected the 50% of negatives located closest to the *imbalanced hyperplane* and then applied random undersampling to remove half of them. Then we randomly oversampled the positives to balance the dataset.

2) 75%-under(0.67)-over

In this method we first selected 75% of negative examples located closest to the *imbalanced hyperplane* and then removed 2/3 of them by random undersampling. Then we applied oversampling to balance the dataset.

3) 100%-under(0.75)-over

In this method we selected all the negatives and removed 75% of them by random undersampling. Then we over-sampled the positives randomly to balance the dataset.

We applied these resampling methods to balance the datasets considered in this research, developed SVM classifiers and evaluated their results by using 5-fold cross-validation. Due to the randomness involved in these resampling methods, each method was repeated five times for each cross validation training partition and results were averaged. The results obtained are compared with the random oversampling results in Table 5. As earlier, we considered the Euclidean distance to compare the results obtained by the proposed resampling methods with the random oversampling results. Here we also present the random undersampling results and the results obtained by the *100%-under(0.5)-over* method (the best 50% data reduction method), which is given in italic type, for the comparative reasons.

TABLE 5. COMPARISON OF THE CLASSIFICATION RESULTS OBTAINED BY THE PROPOSED RESAMPLING METHODS WHICH REDUCE DATA BY 75%

Dataset	Method	Results		
		SE	SP	ED
miRNA	Oversampling	**89.93**	**96.53**	
	Undersampling	91.03	93.02	3.68
	50%-under(0.5)-over	87.99	97.93	2.39
	75%-under(0.67)-over	89.29	96.97	0.77
	100%-under(0.75)-over	**90.05**	**95.85**	**0.69**
	100%-under(0.5)-over	*90.74*	*96.16*	*0.89*
Promoter	Oversampling	**68.89**	**82.56**	
	Undersampling	70.08	80.67	2.23
	50%-under(0.5)-over	62.43	86.30	7.46
	75%-under(0.67)-over	66.26	84.00	2.99
	100%-under(0.75)-over	**68.81**	**81.79**	**0.78**
	100%-under(0.5)-over	*69.23*	*81.13*	*1.46*
Splice-site	Oversampling	**87.88**	**91.12**	
	Undersampling	89.33	89.74	2.00
	50%-under(0.5)-over	82.98	93.83	5.60
	75%-under(0.67)-over	**87.60**	**91.58**	**0.54**
	100%-under(0.75)-over	88.75	90.42	1.12
	100%-under(0.5)-over	*88.75*	*90.83*	*0.92*
Drug-like	Oversampling	**88.29**	**88.94**	
	Undersampling	90.75	84.67	4.93
	50%-under(0.5)-over	84.84	92.67	5.08
	75%-under(0.67)-over	86.34	90.02	2.23
	100%-under(0.75)-over	**88.33**	**88.48**	**0.46**
	100%-under(0.5)-over	*88.29*	*88.44*	*0.50*
Pageblocks	Oversampling	**91.83**	**94.88**	
	Undersampling	93.74	92.52	3.04
	50%-under(0.5)-over	82.61	96.38	9.34
	75%-under(0.67)-over	88.70	95.34	3.16
	100%-under(0.75)-over	**92.70**	**94.23**	**1.09**
	100%-under(0.5)-over	*89.57*	*94.90*	*2.26*

Interestingly, we can observe that the best 75% data

reduction method (depicted in bold type) gave closer results to the random oversampling results than the best 50% data reduction method based on the values of ED. That is, *100%-under(0.75)-over* method resulted in the closest results to the random oversampling results for *miRNA, Promoter, Drug-like* and *Pageblocks* datasets among all the efficient resampling methods considered in this research. Although for the *Splice-site* dataset the *75%-under(0.67)-over* method resulted in the closest results, the results given by the *100%-under(0.75)-over* method were not much far away from the results given by the random oversampling method.

We finally compare the R values (the amount of SP sacrificed when increasing the SE by 1%) calculated with respect to the SE increased and SP decreased by the *100%-under(0.75)-over* method with the R values of the random oversampling and the undersampling methods in Table 6. From these results we can see that although the amount of SP sacrificed by the *100%-under(0.75)-over* method was a bit higher than the random oversampling method, it was lower than the random undersampling method.

TABLE 6. COMPARISON OF THE R VALUES

		dSE	dSP	R
miRNA	Oversampling	+7.15	-2.92	0.41
	Undersampling	+8.25	-6.43	0.78
	100%-under(0.75)-over	+7.27	-3.60	0.49
Promoter	Oversampling	+43.20	-16.89	0.39
	Undersampling	+44.39	-18.78	0.42
	100%-under(0.75)-over	+43.12	-17.66	0.40
Splice-site	Oversampling	+14.32	-5.59	0.39
	Undersampling	+15.77	-6.97	0.44
	100%-under(0.75)-over	+15.19	-6.29	0.41
Drug-like	Oversampling	+16.70	- 8.21	0.49
	Undersampling	+19.26	-12.48	0.65
	100%-under(0.75)-over	+16.74	- 8.67	0.52
Pageblocks	Oversampling	+33.57	-4.63	0.14
	Undersampling	+35.48	-6.99	0.20
	100%-under(0.75)-over	+34.44	-5.29	0.15

VII. CONCLUSION

In this paper, first, we demonstrated that the random oversampling method can increase SE with a lower rate of reduction of SP than the random undersampling method. Then we pointed out that the main problem generated by the random oversampling method as the increase of computational cost due to the addition of new training examples. In order to overcome this problem we experimented with the development of different efficient resampling methods. Through two sets of efficient resampling methods (first reduction of data by 50% and then reduction by 75%) we demonstrated that the comparable classification results to oversampling can be obtained by using fewer amounts of data. In these experiments we observed that what is more important in class imbalance is the imbalance of the distribution of positive and negative data densities around the class boundary.

By considering the experimental results obtained and amount of data reduced, we can recommend that the *100%-under(0.75)-over* method for obtaining comparable classification results to the oversampling results, which generates only 25% of the data compared to the random oversampling method. As future work, it would be interesting to experiment with larger datasets to investigate the possibility of obtaining comparable classification results with oversampling results by further reduction of the training data.

REFERENCES

[1] H. He and E. Garcia. "Learning from imbalanced data," *IEEE Transactions on Knowledge and Data Engineering*, vol.21, no.9, pp.1263-1284, 2009.

[2] G. Weiss. "Mining with rarity: a unifying framework", *SIGKDD Explorations Newsletter*, vol.6, no.1, pp.7-19, 2004.

[3] N. Chawla, N. Japkowicz and A. Kolcz. "Editorial: special issue on learning from imbalanced data sets", *SIGKDD Explorations Newsletter*, vol.6, no.1, pp.1-6, 2004.

[4] J. Showe-Taylor and N. Christianini., "Support vector machines and other kernel-based learning methods", Cambridge University Press, 2000.

[5] K. Veropoulos, C. Campbell and N. Cristianini. "Controlling the sensitivity of support vector machines", in *Proceedings of the International Joint Conference on Artificial Intelligence*, Stockholm, Sweden, 1999, pp.55-60.

[6] R. Akbani, S. Kwek and N. Japkowicz. "Applying support vector machines to imbalanced datasets", in *Proceedings of 15th European Conference on Machine Learning*, Pisa, Italy, 2004, pp.39-50

[7] X.Y. Liu, J. Wu, and Z.H. Zhou, "Exploratory under sampling for class imbalance learning," *Proc. Int'l Conf. Data Mining*, pp. 965-969, 2006

[8] J. Zhang and I. Mani, "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

[9] Burges C. "A tutorial on support vector machines on pattern recognition". *Data Mining and Knowledge Discovery*, 1998, 2(2), 121-167.

[10] Tsang I. et al. "Core vector machines: fast SVM training on very large data sets". *Journal of Machine Learning Research*, 2005, 6, 363-392.

[11] R. Batuwita, V. Palade. "Adjusted geometric-mean: a novel performance measure for imbalanced bioinformatics datasets learning", Submitted to *BMC Bioinformatics*, 2010

[12] A. Asuncion and D. Newman. *UCI repository of machine learning database*. School of Information and Computer Science, University of California, Irvine, CA, 2007. Available: http://www.ics.uci.edu/~mlearn/MLRepository.html.

[13] R. Batuwita and V. Palade. "microPred: Effective classification of pre-miRNAs for human miRNA gene prediction", *Bioinformatics*, vol.25, pp.989-995, 2009.

[14] R. Batuwita and V. Palade, "An improved non-comparative classification method for human miRNA gene prediction". in *Proc. of 8th IEEE Int. Conf. on Bioinf. and Bioneng.,* Athens, Greece, 2008, pp.1-6.

[15] A. Baten et al. "Splice site identification using probabilistic parameters and SVM classification". *BMC Bioiformatics*, 2006, 7(5):S15

[16] H. Xu et al.. "Learning the drug target-likeness of a protein." *Proteomics*, 2007, 7:4255-4263.

[17] C-C. Chang and C-J. Lin. "LIBSVM: a library for support vector machines", 2001. Available http://www.csie.ntu.edu.tw/~cjlin/libsvm