Hybrid probabilistic sampling with random subspace for imbalanced data learning

Peng CAO [a,b,c,1], Dazhe ZHAO [a,b] and Osmar ZAIANE [c]

[a] *College of Information Science and Engineering, Northeastern University, Shenyang, China*

[b] *Key Laboratory of Medical Image Computing of Ministry of Education, Northeastern University, Shenyang, China*

[c] *Computing Science, University of Alberta, Edmonton, Alberta, Canada*

Abstract. Class imbalance is one of the challenging problems for machine learning in many real-world applications. Other issues, such as within-class imbalance and high dimensionality, can exacerbate the problem. We propose a method HPS-DRS that combines two ideas: **H**ybrid **P**robabilistic **S**ampling technique ensemble with **D**iverse **R**andom **S**ubspace to address these issues. HPS improves the performance of traditional re-sampling algorithms with the aid of probability function, since it is not sufficient to simply manipulate the class sizes for imbalanced data with complex distribution. Moreover, DRS ensemble employs the minimum overlapping mechanism to provide diversity and weighted voting, so as to improve the generalization performance. The experimental results demonstrate that our method is efficient for learning from imbalanced data and can achieve better results than state-of-the-art methods for imbalanced data.

Keywords. Classification, class imbalance, sampling method, ensemble learning, random subspace method

1. Introduction

Class imbalance has been recognized as a crucial problem in machine learning and data mining [7,17]. This problem occurs when the training data is not evenly distributed among classes; that is when some classes are significantly larger than others. This problem is growing in importance and has been identified as one of the 10 main challenges of Data Mining [39]. This problem is also especially critical in many real world applications, such as anomaly detection in credit card transaction, fraud detection, medical diagnosis etc. The imbalanced data issue occurs not only in stationary environments, but also in data stream [21,16]. In these cases, standard classifiers generally perform poorly. Classifiers usually tend to be overwhelmed by the majority class and ignore the minority class examples, since most classifiers assume an even distribution of examples among

[1]Corresponding Author: Peng Cao, Neusoft Research Institute, No 2 Xinxiu Street, Shenyang, PR China, 110179. Tel.: +8624 8366 5404; +8624 8366 3446; E-mail: pcao1@ualberta.ca.

classes. Therefore, we need to improve traditional algorithms so as to handle imbalanced data.

Much work has been done in addressing the class imbalance problem. The proposed methods can be grouped in two categories: the data perspective and the algorithm perspective [6,37]. The methods with the data perspective re-balance the class distribution by re-sampling the data space either randomly or deterministically, so that the minority class can be well represented in the training set [17]. Cost-sensitive learning tries to learn more characteristics of samples with the minority class by setting a high cost to the misclassification of samples of the minority class [12,14]. In addition, ensemble methods [9,20,8] are also a good solution for solving the class imbalance problem, since ensemble learning is incorporated with a re-sampling technique or a weighting strategy to acquire better classification performance and generalization capability.

The re-sampling technique is the most straightforward and effective method for dealing with imbalance, since it is not dependent on the classifier and is simple to implement. Weiss et al. observed that the naturally occurring distribution is not always optimal [36]. Therefore, one needs to modify the original data distribution. The existing re-sampling indeed improves the classification performance on the imbalanced data as a whole. Nevertheless, they target only the characteristic of the imbalanced distribution between classes. Besides the between-class imbalance, the presence of inherent complex structures in the data distribution, such as within-class imbalance [23,22] and high-dimensionality [29], are other critical factors of decreasing classification performance. Within-class imbalance refers to the case where a class is formed of a number of sub-clusters with different sizes, concerns itself with the distribution of representative data for subconcepts within a class [23,22]. The existence of within-class imbalance is closely intertwined with the problem of small disjuncts, which has been shown to greatly decrease classification performance [35,24,32]. In addition, high-dimensionality poses further challenges when dealing with class-imbalanced prediction. The complex data distribution aggravates the imbalanced data classification, since the sensitivity of traditional classifiers to class imbalance increases with the domain complexity and the degree of imbalance.

Although traditional re-sampling methods across the entire data distribution can reduce the imbalance between two classes from a global perspective, they cannot solve the within-class imbalance issue as they cannot create appropriate instances without considering the spatial distribution from local distribution, leading to decreased performance. Moreover, existing re-sampling techniques only manipulate the instance in the whole feature space; the irrelevant and redundant features in the high dimensional space may cause the synthetic instances to be inaccurate and biased. To counter the complications introduced above, the solution needs to follow three criteria.

(1) The within-class imbalance may occur in the class distribution. The re-sampling procedure needs to be conducted according to the local distribution by considering the position and distance rather than manipulating data blindly on the whole region. Therefore, the original data need to be decomposed into simple local regions based on clustering, thus the local regions need to be re-sampled separately.

(2) Unlike the scheme duplicating the instances or interpolating new instances along the line between two instances, the artificial instances need to be created through a probability distribution with learned parameters from the training set, so as to generate more accurate and actual instances.

(3) The problem becomes even more severe when imbalanced data sets are involved with high dimensions [29]. The re-sampling may gain more when working in some random reduced feature spaces instead of the whole large feature space, given that it will not only inject more diversity into the scheme of re-sampling, but also reduce the negative influence by the irrelevant and redundant attributes.

These criteria above suggest that when conducting re-sampling, two key issues need to be considered: where the new instances are generated, and how the new instances are produced. The criteria aim to generate more accurate samples which can obey the real sampling, in order to improve the recognition ability of the minority class without a big loss of the prediction ability on the majority class.

The contributions of this work can be listed as follows:

(1) On the basis of the criteria introduced above, we propose a hybrid probabilistic re-sampling algorithm with Gaussian Mixture Model (GMM) [34], which provides an effective solution for re-sampling according to the distribution probability of each instance. It offers an effective solution for within-class in tandem with the between-class imbalance. The procedure of the hybrid re-sampling is carried out according to the distribution probability without jeopardizing structure of data.

(2) To avoid the impact on the clustering of GMM and procedure of re-sampling due to some irrelevant features in the whole feature set, as well as to improve the classification performance, we design a novel ensemble based on the Diverse Random Subspace (DRS). When constructing the random subspace ensemble [19], we explicitly guarantee a minimum overlap of each component through the generation of diverse subsets and a final weighted average based on the performance of each component.

This paper is organized as follows: After presenting a brief review of within-class imbalance as well as small disjuncts issues, and re-sampling algorithms under class imbalance in Section 2, we introduce in Section 3 our proposed method: HPS-DRS. Section 4 details the experimental results comparing our approach to other methods in the literature.

2. Related work

2.1. Within-class imbalanced and small disjuncts

Within-class imbalance refers to the case where a class is formed of a number of sub-clusters with different sizes, concerns itself with the distribution of representative data for subconcepts within a class [23,22]. This can cause problems as we shall discuss later. Real data is commonly distributed according to a mixture density whose components have relative densities that may vary greatly, since the data from the same class may not be homogeneous or arise from noisy misclassified instances. Hence the data with within-class imbalanced distribution occurs. When faced with such a situation the existing methods that address the class imbalance problem may be counterproductive. While they decrease the difference between the prior probabilities of the classes (the between-class imbalance), there is a chance they will increase the difference between the relative densities of the sub components within each class (the within-class imbalance).

Within-class imbalanced data distribution may yield small disjuncts, which is the essential challenge in the within-class imbalanced data issue. A phenomenon sometimes

referred to as the problem with small disjuncts and that these small disjuncts collectively contribute a significant portion of the total test errors [18]. Weiss suggests that there is a relation between the problem of small disjuncts and class imbalance, stating that one of the reasons why small disjuncts have a higher error rate than large disjuncts is due to between-class imbalance [35]. Japkowicz enhances this hypothesis stating that the problem of learning with class imbalance is increased when it yields small disjuncts [23].

Holte et al. [18] evaluate several strategies for improving learning in the presence of small disjuncts. They show that the strategy of eliminating all small disjuncts is ineffective, because the emancipated examples are then even more likely to be misclassified. The common effective solutions to overcome the small disjuncts as well as within-class imbalance are to design better classifiers that address the problem with small disjuncts, and re-sampling approaches. Ting [33] designed a hybrid method using C4.5 to determine if an example is covered by a small or large disjunct. If it is covered by a large disjunct, then C4.5 is used to classify the example. However, if the example is covered by a small disjunct, then IB1, an instance-based learner, is used to classify the example. Japkowicz proposed a cluster-based oversampling algorithm (CBOS), which solves the between-imbalance and within-imbalance issues at the same time [23]. It makes use of K-means clustering to separate into multiple clusters the instances in each class, then a random oversampling method is used for targeting within-class imbalance in tandem with the between-class imbalance. However, some issues remain:

(1) The number of clusters needs to be fixed by the user. It is difficult to determine this number, and if it is not exact, it can significantly impact the result of re-sampling.

(2) The clustering is done in the whole feature space. Many features are redundant and noisy, potentially leading to inaccurate clustering, resulting in inappropriate re-sampling. This is particularly true in the case of high dimensional datasets. In the case of high dimensionality, it is also inefficient. Moreover, data often exhibit characteristics at a local level rather than the global level; therefore, it is necessary to manipulate the data in the reduced subspace.

(3) The instances chosen to be over-sampled are random, resulting in overfitting [6]. Moreover, there still exit redundant instances in the majority class.

2.2. *Re-sampling methods for imbalanced data learning*

Re-sampling methods are attractive under most imbalanced circumstances. This is because re-sampling adjusts only the original training data set instead of modifying the learning algorithm; therefore it provides a convenient and effective way to deal with imbalanced learning problems using standard classifiers. In this paper, we are only interested in sampling based approaches and hence we provide a brief overview of the methods proposed in this category.

2.2.1. *Over-sampling techniques*

1. SMOTE algorithm

   A popular and effective over-sampling method is the synthetic minority over-sampling technique (SMOTE) [6]. SMOTE introduces synthetic examples throughout the line segments joining all of the k minority class nearest neighbors of each minority class instance. SMOTE can address the between class imbalance issue. However it manipulates the instances blindly without considering

the data distribution on the whole feature space, resulting in creating wrong instances when the class dispersion or class noise exist. There exist many methods based on the SMOTE for generating more appropriate instances. ProWSyn [3] was proposed to create weight values for original minority samples based on sample's proximity information, and MWMOTE proposed in [2] can not only select the hard-to-learn minority class samples effectively but also assigns them weights appropriately.

2. Ensemble over-sampling
   The ensemble framework has also drawn attention in the context of learning from imbalanced data. Not only multiple classifiers could have better answer than a single one, but also the ensemble framework provides diversity for avoiding overfitting. SMOTEBoost [8] and RAMOboost [9] were designed to alter the imbalanced distribution based on Boosting. Hoens and Chawla recently proposed the random subspace method integrating SMOTE and empirically demonstrated that sampling methods, achieve better when working in a reduced feature space [20].

3. Cluster-based over-sampling
   Japkowicz proposed a cluster-based oversampling algorithm (CBOS) with K-means clustering [23]. Wu et al. proposed a local clustering method with over-sampling (COG) for rare class analysis, which performs clustering within each class and produces linearly separable subclasses with relatively balanced sizes[38]. The COG has more impact on the performances of linear classifiers.

2.2.2. Under-sampling techniques

Besides over-sampling methods, under-sampling is a popular method in dealing with class-imbalance problems. Under-sampling uses only a subset of the majority class based on the data characteristics of the majority class. Liu et al. [28] proposed two ensemble methods to strengthen the use of majority class examples, called EasyEnsemble and BalanceCascade. They construct an ensemble of ensembles with Bagging, where each individual classifier is also an ensemble of AdaBoost. Yen et al. [40] proposed a cluster-based under-sampling approach for selecting the representative data from the majority class, so as to improve the classification accuracy for the minority class. Zhang et al. [30] proposed an informed under-sampling using K-nearest neighbor (KNN) classifier to achieve under-sampling. Based on the characteristics of the given majority data distribution, four KNN under-sampling methods were proposed, in order to select a representative subset of the majority class. Yu et al. [41] developed a under-sampling method based on the idea of ant colony optimization (ACO) to filter less informative majority samples and search the corresponding optimal training sample subset.

3. HRS-DRS Method

In this section we begin by describing the hybrid probabilistic sampling. We then describe how to incorporate this method into the diverse random subspace ensemble to create HPS-DRS.

3.1. Hybrid Probabilistic Sampling

Gaussian Mixture Models (GMM) are generative probabilistic models of several Gaussian distributions for density estimation in machine learning applications. A Gaussian mixture can be constructed to acceptably approximate any given density. Therefore, we assume the distribution of two classes follows the Gaussian mixture model with unknown parameters. The parametric probability density function of GMM is defined as a weighted sum of Gaussians. The finite Gaussian mixture model with k components may be written as:

$$p(y|\mu_1,\ldots,\mu_k;\sigma_1,\ldots,\sigma_k;\pi_1,\ldots,\pi_k) = \sum_{j=1}^{k} \pi_j N(\mu_j,\sigma_j) \tag{1}$$

and

$$0 \leq \pi_j \leq 1, \sum_{j=1}^{k} \pi_j = 1 \tag{2}$$

where μ_j are the means, σ_j are covariance matrixes, π_j are the mixing proportions, and $N(\mu_j,\sigma_j)$ is a Gaussian with specified mean and variance.

We need to estimate the parameters of GMM with the existing instances of both the classes. The standard method used to fit finite mixture models to observe data is the expectation-maximization (EM) algorithm, which converges to a maximum likelihood estimate of the mixture parameters. However, the drawbacks are that it is sensitive to initialization and it requires the number of components to be set by users. Since the FJ algorithm [15] tries to overcome the major weaknesses of the basic EM algorithm particularly vis-à-vis the initialization, and can automatically select the number of component, we use it here to estimate the parameters of GMM.

Each instance x_i will then be assigned to the cluster k where it has the largest posterior probability $p(k|x_i)$. When calculating the probability of each instance on each component, the probabilities for the numeric attributes is obtained by a Gaussian density function, and for the nominal attributes, the probabilities of occurrence of each distinct value are determined using Laplace estimates. At the same time, we obtain the parameters of each Gaussian component. For different clusters, the re-sampling rates are different; within the cluster, the probabilities of each instance to be chose for re-sampled are different.

We use the over-sampling combined with under-sampling to balance the class size. The sizes of the two classes are M_{maj} and M_{min}. The gap G between two uneven classes is: $G = M_{maj} - M_{min}$. Thus, the amount of instances in the minority class for over-sampling is: $OS_{min} = G \times \alpha$, and the amount of instances in the majority class for under-sampling is: $US_{maj} = G \times (1-\alpha)$. To adjust the within class imbalance, we need to balance cluster sizes in each class. For the majority class, the numbers of instances to be under-sampled are proportional to the size of the cluster; for the minority class, the number of instances to be over-sampled is inversely proportional to the size of the cluster. For example, there are three clusters of size 20, 15 and 10 in the majority class, and two clusters of size 10 and 5 in the minority class. If α is set to 50%, the gap G is 30, $N_{US} = N_{OS} = 15$. The sizes of the three clusters in the majority class become 13, 10 and 7 after under-sampling, while both the sizes of the two clusters in the minority class become 15 after over-sampling. This reduces the within class imbalance, and in this case equalizes the class sizes.

Furthermore, we use the probabilities of each instance to conduct the re-sampling with maintaining the data structure, in order to address the two type imbalance issues. In the clusters of the majority class, the instances with higher probability are dense, they are frequent in the subclass, and hence they have higher chance to be under-sampled. We choose the instances to be under-sampled according to the Gaussian distribution. In the clusters of the minority class, the new instances are produced according to the probability function of Gaussian distribution, resulting in finding more potentially interesting regions. The main steps in under-sampling for the clusters of the majority class and over-sampling for the clusters of the minority class according to the distribution probability are the following:

Over-sampling phase:

Step 1:

In the over-sampling for the minority class, the smaller the size of cluster within the class, the more instances are over-sampled, so as to avoid the small disjuncts. For the i-th cluster, the amount of synthetic instances needed to be generated is:

$$N_{OS}^{i} = \left(\frac{1}{size_{min}^{i}} \Big/ \sum_{j=1}^{N_{min}} \frac{1}{size_{min}^{j}} \right) \times N_{OS} \tag{3}$$

where $size_{min}^{j}$ is the size of i-th cluster, N_{min} is the number of clusters in the minority class.

Step 2:

In the i-th cluster, N_{OS}^{i} instances are generated with the parameters from the current Gaussian distribution. The new instances are generated according to the probability function of the Gaussian distribution with parameters learned from the available data. Firstly, the probability from the Gaussian distribution of each instance is calculated and normalized:

$$\hat{p}_k = p_k \Big/ \sum_{j=1}^{size_{min}^{i}} p_j \tag{4}$$

Then, the amount of new instances for each instance x_k is obtained according to:

$$n_k = \hat{p}_k \times N_{OS}^{i} \tag{5}$$

For ensuring that synthetic instances created via this method always lay in the region near x_k, the n_k instances are generated in its K nearest neighbors region. It can extend more potential regions rather than being limited along the line between the minority example and its selected nearest neighbors. In addition, this guarantees the creation of minority samples in the cluster, and avoids any incorrect synthetic instance generation.

Under-sampling phase:

Step 1:

In the under-sampling for the majority class, we calculate the amount of instances to be under-sampled for each cluster. The number of instances to be under-sampled are proportional to the size of clusters. For the i-th cluster, the amount of instances needed to be removed is:

$$N_{US}^{i} = \left(size_{maj}^{i} \Big/ \sum_{j=1}^{N_{maj}} size_{maj}^{j} \right) \times N_{US} \tag{6}$$

where $size_{maj}^{i}$ is the size of i-th cluster, N_{maj} is the number of clusters in the majority class.

Step 2:

In each component Gaussian distribution, the center region is denser than the border region. These instances from the center are more possible to be redundant, and so are better candidates to be under-sampled. We need to choose the instances to be ignored or removed located on the center of the distribution more than the border. The probabilities to be chosen for under-sampling are proportional to the normalized probability $\hat{p}$ of the Gaussian distribution for each instance in a cluster.

Before applying GMM, to avoid the effect of noise instances, we filter out the noise by checking the labels of nearest neighbors. We remove any noisy example which violates the rule that the class label of each instance is consistent with the one of at least three of its five nearest neighbors. The rule extends the Edited Nearest Neighbor (ENN) Rule [37], which is an under-sampling method that removes data examples whose class label differs from that of at least two of its three nearest neighbors. ENN could inadvertently remove important points when just two out of three neighbors are labeled differently. For example in Figure 1, point A is critical as it locates the decision region, however it would be removed if only three neighbors (solid line) are considered according to the ENN rule, resulting in loss of important information. It is remained when the region of nearest neighbors is expanded to five (dashed line). Therefore, in order to determine the role of a instance more accurately and globally, the size of the nearest neighbors is expanded from three to five. If more than half of its nearest neighbors belong to the opposite class, it is regarded as noise and removed.

The procedure described above is the main scheme of HPS. This general idea of HPS and the difference of this idea to SMOTE are visualized in Figure 2. The (a) is the original skewed data distribution. We can see the minority class has two subclasses with within-class imbalance and an outlier instance. These factors may decrease effectiveness of the learning and over-sampling. The (b) is the result of the SMOTE. The procedure of SMOTE conducts the linear interpolation between nearest neighbor instances, resulting in generating many wrong minority instances under the complex distribution. We see that, some wrong minority samples are interpolated into the region of the majority class since noise and class dispersion exist. Hence, it is not sufficient to manipulate the class size without considering the local distribution. The (c) and (d) show the strategy of our HPS. The clustering result of GMM is shown in (c). (d) is the final result of the HPS. We can see that HPS is able to broaden the decision regions and the concept of minority class from a global perspective to a perspective that encompasses local information in order to deal with within-class imbalance and small disjuncts issues.

3.2. Integration of HPS and Diverse Random Subspace, HPS-DRS

The redundancies and noise in the feature set hinder the re-sampling techniques to achieve their goals. Moreover, the quality of probability estimation and classification

will largely depend on the feature set. The irrelevant or redundant features can lead to a decrease in performance on the re-sampling and prediction.

An important trend in machine learning is the appearance of ensemble learning which combines the decisions of multiple weak classifiers to form an integrated output, so as to provide a diversity for avoiding the overfitting for some algorithms. Moreover, ensemble learning is also a good solution for solving the class imbalance problem, as it is incorporated with re-sampling technique to acquire better classification performance and generalization capability. Ho showed that the random subspace method is able to improve the generalization error [19]. In the random subspace ensemble, the individual classifier is built by randomly projecting the original data into subspaces and training a proper base learner on these subspaces to capture possible patterns that are informative on classification. The majority voting scheme is utilized when combining each specific classifier's prediction.

Under the current standard random subspace scheme, there are three disadvantages requiring improvement: 1) it only picks the feature subset for the original feature set randomly without considering the diversity of instances. Projecting the feature space on a given subspace could produce or enhance noisy instances and even contradicting instances that would lead to poor performance. This is the case when values of attributes in the selected subspace are outliers; 2) Since the features for a classifier are selected independently from the feature subspaces of other classifiers in the ensemble, the standard RS scheme has random characteristics through the selection of feature subsets. However, there are still strong overlaps of the instances with feature selected when constructing individual classifiers on different subspaces, as there is no formulation to guarantee small or reduced overlap; 3) because some subspaces may contain noisy features and individual classifier developed from these subspaces are not informative, it is not correct treating each classifier as if it contributed equally to the group's performance; there is a lack of attention to appropriate weight assignments to individual classifiers according to their respective performance based on the different subspace.

Therefore, we propose an improvement of RS, called DRS (Diverse Random Subspace) for addressing these disadvantages. Firstly, we extend the common random subspace by integrating bootstrapping samples in order to obtain the diversity with respect to instances and features. In the bootstrapping method, different training subsets are generated with uniform random selection with replacement. Secondly, it cannot ensure the diversity of each subset since the instances and the features are chosen randomly without considering previously selected subspaces for other classifiers. Therefore, to improve diversity between each subset, we use a formulation to make sure each subset is diverse. We introduce a concept of *overlapping rate* of subsets:

$$overlapping\ rate = \frac{subset_i \cap subset_j}{N_{fea} \times N_{ins}} \tag{7}$$

where the $subset_i$ and $subset_j$ are two subsets within certain subspaces, N_{fea} and N_{ins} are the feature size and instance size of each subset; e.g., in Figure 3, the overlapping rate is 16%.

We then introduce a threshold T_{over} to control the intersection between each subset. The overlapping rate of all the subsets should be smaller than the threshold T_{over}.

The *GenerateDiverseSets* described in Algorithm 1 generates a diverse set *DiverseSet*, by iteratively projecting bootstrap sample D_k into the specific random subspace $RS(D_k)$. The function $isDiverse(RS(D_k), DiverseSet, T_{over})$ examines if the new projection $RS(D_k)$ is diverse enough from the previously collected projections in *DiverseSet* based on the overlapping region threshold T_{over}. The generation of projections stops when there is stagnation *sr*, after enough trials, no new projection is diverse enough from the collected subsets. It enforces the diversity or independence by minimizing the overlapping region among the subset with subspace used previously.

Algorithm 1 GenerateDiverseSets

Require:

Training Dataset, D_{train}
Ratio of bootstrap samples, R_s
Ratio of feature subspace, R_f
Overlapping region threshold, T_{over}
Stagnation rate, *sr*

Ensure:

Diverse dataSets, *DiverseSets*

$change = 0; DiverseSet = \{\}$
while $change < sr$ **do**
  A bootstrap sample D_k selected with replacement from D_{train} with R_s
  Select an random subspace with R_f from D_k
  if $isDiverse(subspace(D_k), DiverseSet, T_{over}) == true$ **then**
    $DiverseSet.add(subspace(D_k)); change = 0;$
  else
    $change = change + 1;$
  end if
end while

Thirdly, we employ a weighted average while combining classifiers according to the performance of each component. In the diverse subsets, some of the selected subspaces may have better performance on the imbalanced dataset; others lack the ability to properly discriminate between the different classes. We utilized the out-of-bag (OOB) samples in determining different classifier's voting power, and then each base classifier is weighted when combined to create the final decision function. The goal is to assign weights that reflect the relative contribution of each classifier in improving the overall performance of the ensemble. It is known that the use of overall accuracy is not an appropriate evaluation measure for imbalanced data. For example, a dataset for which the majority class represents 99% of the data, and the minority class represents 1% of the data (this dataset is said to have an imbalance ratio of 99:1). In such cases, the classifier which always predicts the majority class will have an accuracy of 99%. When the performance of both classes is concerned, two accuracies of both classes are expected to be high simultaneously. Kubat et al [25] suggested the G-mean defined as the geometric mean of accuracies measured separately on each class: ($G\text{-}mean = \sqrt{ACC_{maj} \times ACC_{min}}$). G-mean measures the balanced performance of a learning algorithm between these two classes, and is commonly utilized when performance of both classes is concerned and expected

to be high simultaneously. Therefore G-mean is chosen to be the metric for representing the performance of each classifier.

The HPS-DRS algorithm is described in Algorithm 2.

Algorithm 2 HPS-DRS

Require:

Training Dataset D_{train}, Test Dataset D_{test}, Ratio of bootstrap samples R_s, Ratio of feature subspace R_f, Overlapping region threshold T_{over}, Hybrid sampling ratio parameter α

Training:

1: $Ensemble = NULL$
2: $DiverseSets$=GenerateDiverseSets(D_{train},R_s,R_f,T_{over})
3: **for** each subset D_k in $DiverseSets$ **do**
4: Apply HPS on the subset D_k, and generate a new balanced set BD_s^k with α
5: Construct a classifier C_k on the BD_k
6: Evaluate C_k on the $OOB(D_k)$ and obtain the value of G-mean, GM_k
7: $C_k.Subspace = Subspace(D_k)$;$C_k.GM = GM_k$
8: **end for**
9: $Ensemble = Ensemble \cup C_k$
10: Calculate and normalize the weights of each classifier in $Ensemble$ according to its GM

Testing:

11: Calculate output from each classifier of $Ensemble$ with D_{test}
12: Generate the final output by aggregating all the outputs with weighted voting

To reduce the learning time of HPS-DRS, the procedure of sampling and learning in each subset D_k can be carried out in parallel before aggregating. Moreover, each classifier is trained in the reduced subset with fewer instances and features. Therefore the computational time is acceptable.

Since data often exhibits characteristics at a local rather than global level, DRS can find more valuable local data properties so as to improve the quality of sampling. Moreover, the different imbalanced data distribution in each random subset makes the ensemble classifier robust to the evolving testing distribution. Furthermore, DRS can alleviate the effect of class overlapping on the imbalanced data distribution [10], since the two classes may be separable in some reduced subspace.

4. Experimental study

4.1. Dataset Description

To evaluate the effectiveness of our method on the classification of different datasets, and to compare with other methods specifically devised for imbalanced data, we tried several datasets from the UCI database containing imbalance (Table 1). Some of these are originally multiple class datasets and were converted into binary class problems by keeping the smallest class as minority and the rest as majority. The datasets used contain different degrees of imbalance from 4% up to an almost balance at 47%.

4.2. Experiment 1: Evaluating the effectiveness of HPS-DRS

In this experiment, we evaluate the effectiveness of our proposed algorithm HPS-DRS. We conduct the comparison between the basic classifier without re-sampling (basic), HPS, HPS-DRS, as well as CHS working on the original RSM framework (HPS-RS). We chose unpruned decision tree (C4.5) with Laplace-smoothing as our base classifier, because it is the most commonly used classifier with sampling for imbalanced datasets in the literature. The ensemble size of all the ensemble methods is set to 50. In the parameters setting of HPS, the α is set to 70% for avoiding loss of reducing too many instances, and the parameters K is set to 5. In the parameters setting of DRS, R_s is 0.7, R_f is 0.5. The best value of T_{over} can be obtained from the training data, then the ensemble size can be determined adaptively. In this experiment, it is set to 0.4 empirically. It is a good trade-off value between the diversity and the sufficient ensemble size according to experiments. Although it is not necessarily the best, it can guarantee the diversity among each subset. All the experiments are carried out by 10-fold cross-validation. The results are shown in Table 2.

From these experiments, we can show that HPS is a good re-sampling method, since it improves C4.5 categorically on all the datasets except Glass. In addition, we can also point out that DRS is a good ensemble framework which can inject more diversity into the bias of sampling and learning algorithm, so as to help HPS acquire better over-sampling performance and achieve a better generalization ability. Especially for the datasets with high dimensional features such as Spambase and Sonar, HPS-DRS achieves good generalization and avoids the negative impact of high dimensionality as well as the strong bias of noisy features; the diverse random subspace method emphasizes ensemble diversity explicitly during training and fuses all the components with weighted voting achieving better performance than the traditional random subspace ensemble on the imbalanced data. This indicates that the diversity in the ensemble can facilitate class imbalance learning. However, our method cannot achieve the best result on some low dimensional datasets, such as Transfusion with only 4 attributes. That is because the random subspace method is more effective when datasets have a large number of attributes.

From the results, we can also notice that HPS-RS and HPS-DRS are performing well and improve upon the basic classifier in the case of a balanced dataset such as Sonar, while the improvement by single HPS is not evident. This is because in the almost balanced data, the value of G is so small ($G = M_{maj} - M_{min}$) and the re-sampling does not change the distribution much. However, there may still exist within-class imbalance. Since HPS is bound by G the effect of HPS is indeed minimal when we have balanced classes. In the case of Sonar it was the DRS that improved upon basic thanks to the ensemble with diverse subspaces. Sonar still has within-class imbalance based on GMM. By lifting the G restriction on HPS, one could further improve the results.

The scarcity in the minority class includes relative scarcity and absolute scarcity. Relative scarcity is when despite the imbalance the minority class is still well represented. Absolute scarcity is when the minority class does not have enough instances to represent it. As HPS is based on the estimation of data distribution, when the amount of instances in the minority is not sufficient, it cannot obtain the exact parameters of data distribution, resulting in low quality of hybrid sampling, such as Glass dataset with only 9 instances. Although the Letter dataset contains the same imbalance level as Glass, it has enough instances in the minority class to be represented in the learned model. There-

fore, HPS based methods can improve C4.5 on Letter dataset effectively. For the absolute scarcity issue, it is still a critical research question that is not well addressed by current approaches. The minority class is so weakly represented that very little can be learned from its rare instances.

4.3. Experiment 2: Comparison between HPS-DRS and state of the art methods

We compare between our method HPS-DRS, and the state-of-the-art methods, such as MetaCost (MC) [12] , SMOTE over-sampling (SM) [6], SMOTEBoost over-sampling (SMB)[8], ENN [37], Cluster-based Over-Sampling (CBOS) [23] and COG with random over-sampling(COG-OS)[38]. These methods were considered because they are commonly used in research on class imbalance, some from the algorithm perspective and some from the re-sampling perspective. Moreover, SMB and MC are the methods integrating the ensemble framework, while CBOS and COG-OS are both based on the clustering techniques for splitting the data into segments. We don't use the non-heuristic random re-sampling in our comparison since that they have potential drawbacks such as information loss or causing overfitting [6,17].

All the SMOTE based over-sampling methods are set to the commonly used 200% threshold. Moreover, the number of nearest neighbors is set to five when generating new synthetic instances as done in the literature. The ensemble sizes in all the ensemble classification method are set to 50 in our experiments. ENN does not require a user specified under-sampling ratio, and K is set to the default value 3 [37]. The ratio cost of the MC is set to the reverse of the sizes of two classes, $RatioCost = N_{maj}/N_{min}$. In CBOS, the value of K in the K-means clustering for each class is set to 2. In COG-OS, the cluster number is set to be 4, and the minority class is over-sampled to the average size of the partitioned large class approximately.

To make our comparisons more convincing, we further use AUC as the performance evaluation. AUC measures the performance of ranking the minority examples above the majority example. It can capture the trade-off between true positives and false positives, producing a robust metric for even the most imbalanced datasets.

From Table 3 and 4, we find that, HPS-DRS provides the best results in terms of G-mean and AUC in most of the datasets. SMOTE and SMB consider the class skewness and properties of the dataset as a whole, and manipulate the instances blindly without taking the majority class into consideration, resulting in overgeneralization [31]. It leads to the creation of wrong instances when the class dispersion or the class noise exists, decreasing the value of G-mean and AUC. Although CBOS solves both imbalance issues, the disadvantages of CBOS introduced in 2.3 lead to poor results. COG-OS is only effective on the linear classifier, thus it cannot offer consistent performance based on the decision tree.

HPS-DRS shows appealing performances on imbalanced data through hybrid sampling with probability function and the diverse ensemble framework. HPS-DRS solves the issues of two types of imbalance more flexibly and efficiently. From Table 3 and Table 4, we can also notice that for the datasets with high dimensional features such as Spambase and Sonar, HPS-DRS offers a great advantage over other sampling techniques.

4.4. Experiment 3: The effectiveness of sampling ratio α on HPS-DRS

The optimal re-sampling ratio is usually unknown. In order to observe the influence of the re-sampling ratio in HPS-DRS on the classification performance, we chose *German* dataset with a moderate imbalanced level 30% as an example of the variation of performance with the ranging of α. The range of α is set to be [0, 1] and the step is set to 0.1. With each α, we conduct a 10-fold cross validation to obtain an averaged G-mean and AUC results.

From Figure 4, we can see the changes of G-mean and AUC when varying the value of α in HPS-DRS. When α is 0, only under-sampling for the majority class is carried out and no new instances are generated. Important information of the majority class may be lost, hence the performance is lowest. When increasing the value of α, the two performances increase. When α is 1, over-sampling for minority class is performed without removing any redundant instance from the majority class. The issue of overfitting may occur due to the large amount of the minority class as well as the redundant information of majority class. Moreover, we found G-mean and AUC to be highest when α is 60% and 65% respectively. It demonstrates the hybrid re-sampling scheme with an appropriate sampling ratio can achieve optimal classification performance. Moreover, it illustrates the effectiveness of the hybrid sampling method compared with each individual re-sampling technique.

Clearly, the choice of α affects directly the final performance, so it is desirable to obtain the optimal parameter of sampling ratio. However, many studies have shown that for certain imbalanced data sets, the prior degree of imbalance between class distribution is not the only factor influencing performance of classification of imbalanced data. Data set complexity (overlapping, lack of representative data as well as small disjuncts), the specific data distribution of each class and the choice of classifier are all factors of final classification performance. For instance, in a dataset with a high between-class imbalanced ratio, if a minority class with a simple concept has sufficient instances, it may not require extra synthetic instances, so the corresponding value of α may be a lower value. If a minority class lacks instances to represent its own concept, the corresponding value of α may be a higher value. Furthermore, the value of α depends on the redundancy level of the majority class. Therefore, there are no explicit relationships between the class prior ratio and the optimal sampling ratio. As a result, we are unable to obtain the optimal sampling ratio beforehand and have to empirically discover the optimal value of the sampling ratio parameter for obtaining the best performance on each dataset.

In order to estimate the optimal parameter α, the best α is chosen by cross validation in the data set. In imbalanced data cases, available data instances, mainly instances of the minority classes, are insufficient for traditional cross validation in the training set. For this reason, we randomly divided the original data set into two sets: the training set (80%) and the validation set (20%) for measuring the performance of each value of α. This process is repeated 10 times. The output is a value of α which yields the best measure metric value among all tests. We chose G-mean and AUC as the guidance metric to tune the parameter, respectively. Moreover, we compared the optimized HPS-DRS with SMB of which over-sampling percentage is optimized by G-mean in the[100%, 500%] range with 50% step. The parameter of sampling level of both ensemble based sampling need to be searched separately. This ensured that both approaches were independently optimized in the same fashion to achieve their best performances.

After identifying the optimal parameter of sampling level for both sampling methods on each dataset, we compared them having this selected sampling parameter. All the methods are conducted by 10 fold cross validation. The section of the 10 fold cross validation is totally independent from the one of cross validation for obtaining the optimal sampling parameter. All the results of G-mean are shown in Table 5. Moreover, we list the optimal sampling parameter value. In the majority of cases, for HPS-DRS, the G-mean value from the G-mean wrapper is higher than the one of the AUC wrapper, but in some cases, the G-mean value from the AUC wrapper is higher, such as Vowel and Spambase datasets. From this, we believe that by using AUC as the wrapper evaluation function we get better performances. Moreover, the optimized HPS-DRS outperform the optimized SMB with the same metric of G-mean on 8 datasets.

4.5. Experiment 4: The robustness of noise effective

Data are said to be noisy if they contain erroneous data values. These erroneous values can occur in the dependent (class noise) or independent (attribute noise) variables in a data set [43]. Noise in imbalanced datasets may exhibit unpredictably negative effects on the performance of learning algorithms. In order to systematically investigate the robustness of HRS-DRS, we manually introduced noise with different levels into the *German* dataset, then assessed the performance of HRS-DRS and other comparative methods on the new datasets. We generated two types of noisy instances manually: class label noisy instances and attribute noisy instances, so as to simulate and evaluate the robustness of our method in both cases.

We employ the same experimental procedure as in [1,9] to inject the label noisy instances into the *German* dataset. Given a noise label l, each instance sees its label permutated with a probability of $l\%$. Table 6 shows the AUC value of HRS-DRS and other comparative learning algorithms under different class label noise levels. Moreover, we employ the same experimental procedure as in [42,9] to generate the attribute noisy instances. Given a noise level l, each attribute in each instance is changed with a different value with a probability of $l\%$. This new value is selected uniformly among the other possible values for this attribute. Table 7 shows the AUC value of HRS-DRS and other comparative learning algorithms under different attribute noise levels. We find that the class noise is more detrimental to classification performance. The results clearly indicate the superiority of our method in the presence of noisy data, particularly the higher the noise level gets.

In our method, the random subspace ensemble decreases the influence of noise in the attribute values while bootstrap sampling reduces the effect of noise in labels. It is possible that noise in the training set do not show up in some certain subset, as the instances with noisy label may not be selected by the bootstrap sampling or the noise attributes may not be chosen by the pseudo-random selection. Moreover, in the procedure of DRS, some instances with noisy label matching the nearest neighbors rule can be removed so as to avoid the impact on the estimation of the probability distribution and the subsequent sampling to some extent.

SMB achieves the best AUC on the original *German* dataset. However, Boosting is less robust in noisy settings. This is expected because noisy examples tend to be misclassified, and the weight will increase for these instances, so SMB has a lower performance as the noise level increases. In addition, SMOTE is sensitive to noise since

the interpolation of new instances is generated along the line between nearest neighbors with the same class label in the feature space. All simulation results presented in this section illustrate the robustness of HRS-DRS to the negative influence of noise.

4.6. Experiment 5: Lung nodule candidate data classification

In the process of lung cancer diagnosis pulmonary nodules are first uncovered. Computer Aided Diagnosis (CAD) systems to detect lung nodule in chest radiographic images ca be broadly divided into two major steps, an initial nodule identification step and a false-positive reduction step [26]. For finding the suspicious nodule candidates, the initial detection of the CAD requires high sensitivity, and so, it produces a number of false positives. The purpose of false-positive reduction is to remove these false positives (FPs) as much as possible while retaining a relatively high sensitivity. This is known as the False Positive Reduction (FPR) problem, which is a binary classification between the nodules (TPs) and non-nodules (FPs). The most significant problem in the FPR is that the two classes are skewed and have unequal misclassification costs.

Class imbalanced data has detrimental effects on the performance of conventional classifiers, resulting in lowering the performance of discrimination in the candidate nodule. However, in nodule classification, the problem has attracted less attention. Only few papers have been published addressing this problem. The authors in [4] use Tomek links with SVM (TL-SVM)to remove borderline false nodule cases in order to achieve 100% sensitivity . Campadelli et al. prove that cost-sensitive SVM (CS-SVM) trained with imbalanced data sets achieve promising results in terms of sensitivity and specificity, by means of adjusting the misclassifications cost of false positives versus false negatives [5]. Dolejsi et al. use asymmetric Adaboost (Asy-Adaboost)learning to improve the sensitivity by setting different weights for two classes [11]. These are the three contenders against which we will compare our approach.

Constructing an accurate classification method requires a training data set that represents different aspects of nodule features. Our feature extraction process generated 43 image features, features that are commonly used in medical image processing for characterizing lesions. Using these features, we constructed the input space for the compared classifiers.

Our database consists of 98 thin section CT scans containing 106 solid nodules, obtained from the Guangzhou hospital in China. These databases included nodules of different sizes (3-30mm). We obtained the appropriate candidate nodule samples using a candidate nodule detection algorithm, which identifies 95 true nodules as positive class and 592 non-nodules as negative class from the total CT scans. Figure 5 shows an example result images of candidate nodule detection.

We chose SVM as the classifier, as it is the most commonly used classification model in the nodule recognition. The intrinsic parameters (C and γ) are obtained by grid search under the guidance of G-mean prior to be combined with sampling or cost strategy. Experimental results in Table 8 show that HPS-DRS improved the performance of nodule recognition as compared to the other three methods. It means that our method can be applied on the nodule or other lesion detection in medical images and improve upon state of the art.

5. Conclusion

In this paper, we presented a probabilistic local over-sampling method, combined with diverse random subspace ensemble for classification on two-class imbalanced data. The class imbalance problem may not be a problem in itself. Rather, the complex distribution such as within-class imbalance and high dimension are responsible for the performance decrease. The main idea of the method is adding more accurate synthetic instances for the minority class on local regions according to the probability function for targeting within-class imbalance in tandem with the between-class imbalance. In addition, the diverse random subspace provides a diverse framework to reduce the affect of high dimension and enhance the generalization ability. We showed that HPS-DRS can achieve better results than state-of-the-art methods for imbalanced data through extensive experiments on multiple benchmark datasets from UCI and a real world dataset regarding lung nodule detection.

As a new method for imbalanced learning problems, there are several interesting future research directions for HPS-DRS. In this paper, HPS-DRS is only evaluated on imbalanced binary class learning; we will extend it to multi-class for improving its applicability in our future research. Moreover, the two ratio parameters (R_s and R_f) in DRS dependent on the data distribution are determined empirically. We will explore a method to determine them automatically according to the data distribution. Furthermore, we will evaluate our method on text data with much higher dimensionality. Two main issues in text data classification are the high dimensional feature space and high feature-to-instance ratio, a classifier usually suffers from the 'curse of dimensionality'. Random subspace ensemble can overcome these issues very well [13] and we can use a multinomial mixture distribution in place of the Gaussian mixture distribution to model and estimate the probability of text data [27] since the bag of words is the typical representation used in text mining. In addition, since HPS is contingent on G, the differential between the majority size and the minority size, we intend to change the formulation within HPS not to be restricted on G, as completely balanced distribution is not always the best distribution.

6. Acknowledgments

This work is supported by the Alberta Innovates Centre for Machine Learning as well as the Chinese National Natural Science Foundation under grant (No. 61172002 and 61001047). Moreover, one author was supported by the China Scholarship Council for two years at the University of Alberta.

References

[1] D. Anyfantis, M. Karagiannopoulos, S. Kotsiantis, and P. Pintelas, Robustness of learning techniques in handling class noise in imbalanced datasets, In *Proceedings of the 4th IFIP International Conference on Artificial Intelligence Applications and Innovations (AIAI 2007)*, Springer, vol. 247, 2007, pp. 21-28.

[2] S. Barua, M.M. Islam, X. Yao and K. Murase, MWMOTE - Majority Weighted Minority Oversampling Technique for Imbalanced Data Set Learning, *IEEE Transactions on Knowledge and Data Engineering*, 2012.

[3] S. Barua, M.M. Islam, K. Murase, ProWSyn: Proximity Weighted Synthetic Oversampling Technique for Imbalanced Data Set Learning, In *Proceedings of the 17th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2013)*, Gold Coast, Australia, 2013, pp. 488–499.

[4] L. Boroczky, L. Zhao, and K. Lee, Feature subset selection for improving the performance of false positive reduction in lung nodule CAD, *IEEE Transactions on Information Technology in Biomedicine*, **10**(3) (2006), 504–511.

[5] P. Campadelli, E. Casiraghi, and G. Valentini, Support vector machines for candidate nodules classification, *Neurocomputing*,**68** (2005), 281–288.

[6] N. Chawla, K. Bowyer, L. Hall, and W. Kegelmeyer, SMOTE: synthetic minority over-sampling technique, *Journal of Artificial Intelligence Research*, 2002, pp. 341–378.

[7] N. Chawla, N. Japkowicz, and A. Kotcz, Editorial: special issue on learning from imbalanced data sets, *ACM SIGKDD Explorations Newsletter*, **6**(1) (2004), 1–6.

[8] N. Chawla, A. Lazarevic, L. Hall, and K. Bowyer, SMOTEBoost: Improving prediction of the minority class in Boosting, In *Proceedings of 7th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD 2003)*,Springer Berlin Heidelberg, Cavtat-Dubrovnik, Croatia, 2003, pp. 107–119.

[9] S. Chen, H. He, and E. A. Garcia, RAMOBoost: Ranked minority oversampling in Boosting, *IEEE Transactions on Neural Networks*, **21**(10) (2010), 1624–1642.

[10] M. Denil and T. Trappenberg, Overlap versus imbalance, *Advances in Artificial Intelligence*, 2010, pp. 220–231.

[11] M. Dolejsi, J. Kybic, S. Tuma, and M. Polovincák, Reducing false positive responses in lung nodule detector system by asymmetric Adaboost, In *Proceedings of 5th IEEE International Symposium on Biomedical Imaging*, IEEE Press, Paris, 2008, pp. 656–659.

[12] P. Domingos, Metacost: a general method for making classifiers cost-sensitive, In *Proceedings of the 5th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD 99)*, ACM, New York, NY, USA, 1999, pp. 155–164.

[13] M.J. Gangeh, M.S. Kamel and R.P.W. Duin, Random subspace method in text categorization, In *Proceedings of the 20th International Conference on Pattern Recognition (ICPR 2010)*, IEEE, Washington, DC, USA, 2010, pp. 2049–2052.

[14] W. Fan, S. Stolfo, J. Zhang, and P. Chan, Adacost: misclassification cost-sensitive Boosting, In *Proceedings of the 6th International Conference on Machine Learning (ICML 99)*, ACM, San Francisco, CA, USA, 1999, pp. 97–105.

[15] M. Figueiredo and A. Jain, Unsupervised learning of finite mixture models, *IEEE Transactions on Pattern Analysis and Machine Intelligence* , **24**(3) (2002), 381–396.

[16] J. Gao, W. Fan, J. Han, and P. S. Yu, A general framework for mining concept-drifting data streams with skewed distributions, In *Proceedings of the 7th SIAM International Conference on Data Mining*, Philadelphia, PA, USA, SIAM, 2007, vol.7, pp. 3–14.

[17] H. He and E. Garcia, Learning from imbalanced data, *IEEE Transactions on Knowledge and Data Engineering* , **21**(9) (2009), 1263–1284.

[18] R.C. Holte, L.E. Acker, and B.W. Porter, Concept learning and the problem of small disjuncts, In *Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI 89)*, 1989, pp. 813–818.

[19] T. Ho, The random subspace method for constructing decision forests, *IEEE Transactions on Pattern Analysis and Machine Intelligence* , **20**(8) (1998), 832–844.

[20] T. Hoens and N. Chawla, Generating diverse ensembles to counter the problem of class imbalance, In *Proceedings of the 14th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2010)*, Hyderabad, India, 2010, pp. 488–499.

[21] T. R. Hoens and N. V. Chawla, Learning in non-stationary environments with class imbalance, In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data*

mining, New York, NY, USA, ACM, 2012, pp. 168–176.

[22] N. Japkowicz, Concept-learning in the presence of between-class and within-class imbalances, In *Proceedings of the 14th Biennial Conference of the Canadian Society for Computational Studies of Intelligence*, Ottawa, Canada ,Springer Berlin Heidelberg, 2001, pp. 67–77.

[23] N. Japkowicz, Class imbalances: are we focusing on the right issue, In *Workshop on Learning from Imbalanced Data Sets*, 2003, vol. 1723, pp. 63–69.

[24] T. Jo and N. Japkowicz, Class imbalances versus small disjuncts, *A*CM SIGKDD Explorations Newsletter, **6**(1) (2004), 40–49.

[25] M. Kubat and S. Matwin, Addressing the curse of imbalanced training sets: one-sided selection, In *Proceedings of the 4th International Conference on Machine Learning (ICML 97)*, Morgan Kaufmann, San Francisco, USA, 1997, pp. 179–186.

[26] Q. Li, Recent progress in computer-aided diagnosis of lung nodules on thin-section CT, *Computerized medical imaging and graphics*, **31**(1) (2007), 248–257.

[27] A. McCallumn and K. Nigam, A comparison of event models for naive bayes text classification, In *AAAI-98 workshop on learning for text categorization*, 1998, vol. 752, pp. 41–48.

[28] X. Liu, J. Wu, and Z. Zhou, Exploratory undersampling for class-imbalance learning, *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* , **39**(2) (2009), 539–550.

[29] R. Blagus and L. Lusa, Class prediction for high-dimensional class-imbalanced data, *BMC bioinformatics* , **11**(1) (2010), 523–539.

[30] A. Liu, J. Ghosh, and C. Martin, kNN approach to unbalanced data distributions: a case study involving information extraction, In *Proceedings of Workshop on Learning from Imbalanced Datasets*, 2003.

[31] B. X. Wang and N. Japkowicz, Imbalanced data set learning with synthetic samples, In *Proceedings of IRIS Machine Learning Workshop*, Ottawa, Canada, 2004.

[32] R.C. Prati, G.E.A.P.A. Batista, and M.C. Monard, Learning with class skews and small disjuncts, In *Advances in Artificial Intelligence–SBIA*, 2004, pp. 296–306.

[33] K.Ming. Ting, The problem of small disjuncts: Its remedy in decision trees, In *Proceedings of the 10th Canadian Conference on Artificial Intelligence*, 1997, pp. 91–97.

[34] D.M. Titterington, A.F.M. Smith, U.E. Makov, *Statistical analysis of finite mixture distributions*, vol. 7. Wiley New York, 1985.

[35] G. Weiss, The impact of small disjuncts on classifier learning, *Data Mining* , 2010, vol. 8, pp. 193–226.

[36] G. Weiss, K. McCarthy, and B. Zabar, Cost-sensitive learning vs. sampling: Which is best for handling unbalanced classes with unequal error costs, In *Proceedings of international conference on data mining (ICDM 07)*, Las Vegas, NV, USA, CSREA Press, 2007, pp. 35–41.

[37] D. Wilson, Asymptotic properties of nearest neighbor rules using edited data, *IEEE Transactions on Systems, Man and Cybernetics*, **SMC-2**(3) (1972), 408–421.

[38] J. Wu, H. Xiong, P. Wu, and J. Chen, Local decomposition for rare class analysis, In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD 07)*, New York, NY, USA, ACM, 2007, pp. 814–823.

[39] Q. Yang and X. Wu, 10 challenging problems in data mining research, *International Journal of Information Technology & Decision Making*, **5**(4) (2006), 597–604.

[40] S.J. Yen, Cluster-based under-sampling approaches for imbalanced data distributions, *Expert Systems with Applications*, **36**(3) (2009), 5718–5727.

[41] H. Yu, J. Ni, and J. Zhao, ACOSampling: An ant colony optimization-based undersampling method for classifying imbalanced DNA microarray data, *Neurocomputing*, vol. 36, 2012, pp. 309–318.

[42] X. Zhu, X. Wu, and Y. Yang Error detection and impact sensitive instance ranking in noisy datasets, In *Proceedings of the American Association for Artificial Intelligence*, Menlo Park, CA, Cambridge, MA: MIT Press, 2004, pp. 378383.

[43] X. Zhu and X. Wu Class noise vs. attribute noise: A quantitative study, In *Artificial Intelligence Review*, **22**(3) (2004), 177-210.

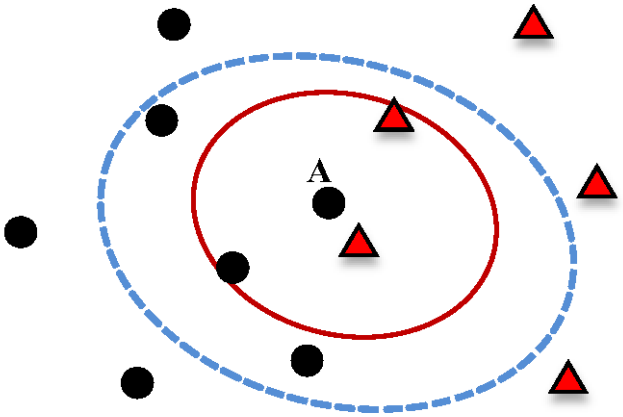

Figure 1. The example of the extended ENN rule. With 3 nearest neighbors, point A would be removed (solid line). It is not removed with 5NN (dashed line)

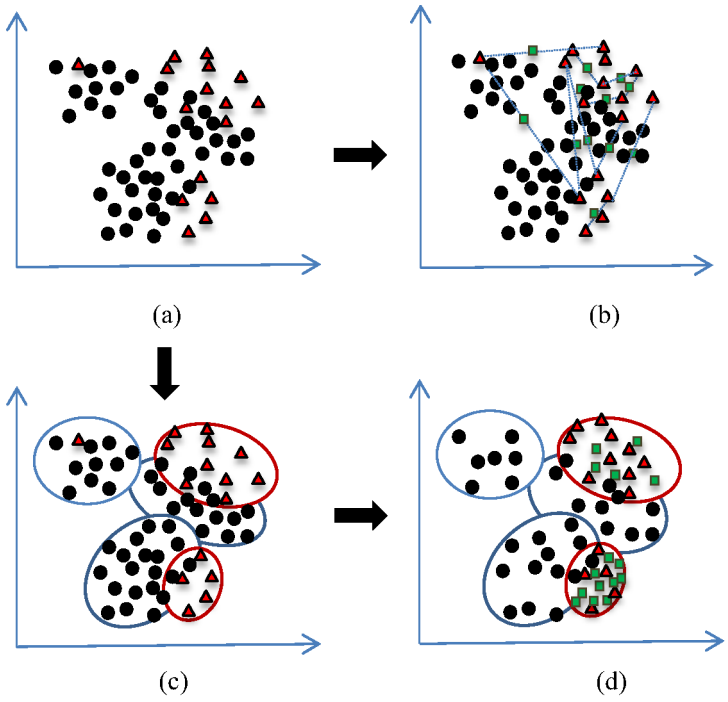

Figure 2. Comparison of different synthetic data generation mechanisms. (The black circles and the red triangles represent the majority and minority classes, respectively. The green squares are the new instances generated by over-sampling). (a) Original imbalanced data distribution. (b) Data distribution after SMOTE. (c) the result of Gaussian mixture clustering. (d) Data distribution after HPS.

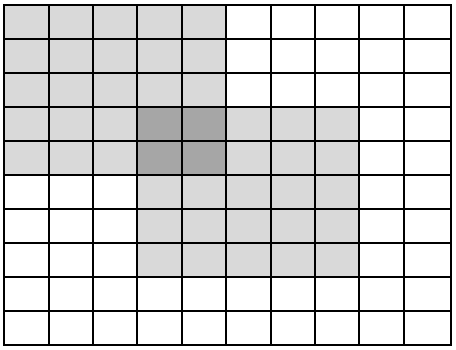

Figure 3. The overlapping rate between two subsets

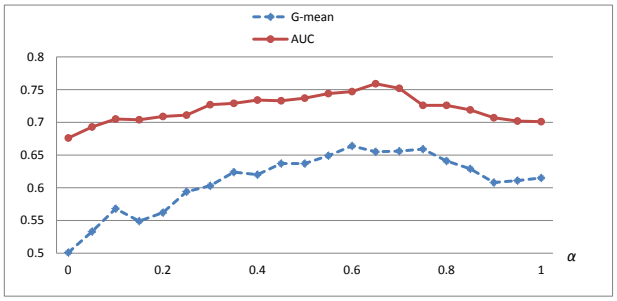

Figure 4. The performance of HPS-DRS in terms of G-mean on *German* dataset

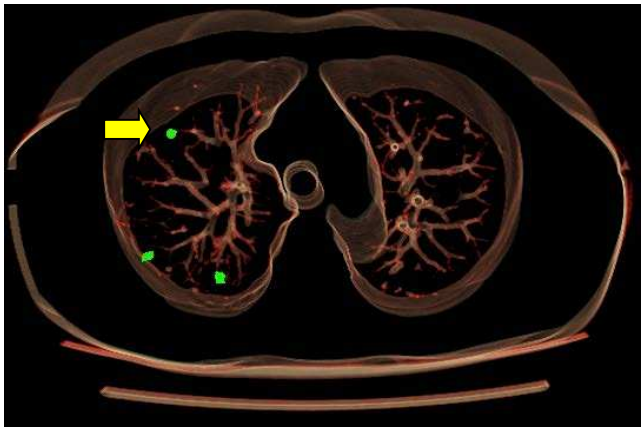

Figure 5. Initial detection result of candidate nodules. TPs indicated by arrow, other spots are FPs

Table 1. Dataset characteristics used in experiments

Dataset(+)	Instances	Features	Class ratio
Glass(tableware)	214	9	4%
Letter(Z)	20000	16	4%
Vowel(hid)	990	10	9%
Page(2,3,4,5)	5473	10	10%
Satimage(damp)	6435	36	10%
Segment(bricface)	2310	19	14%
Transfusion(yes)	748	4	24%
Vehicle(opel)	846	18	25%
German(1)	1000	20	30%
Pima(positive)	768	8	35%
Spambase(spam)	4601	57	40%
Sonar(Rock)	208	60	47%

Table 2. Average values of G-mean for four methods on multiple data sets shown in Table 1.

Dataset	Basic	HPS	HPS-RS	HPS-DRS
Glass	0.859	0.855	**0.863**	0.859
Letter	0.928	0.943	0.965	**0.967**
Vowel	0.982	**0.991**	0.984	0.987
Page	0.755	0.779	0.789	**0.803**
Satimage	0.808	0.831	0.834	**0.846**
Segment	0.933	0.979	0.989	**0.996**
Transfusion	0.571	**0.589**	0.573	0.567
Vehicle	0.607	0.685	0.692	**0.724**
German	0.577	0.628	0.649	**0.656**
Pima	0.612	0.657	0.675	**0.692**
Spambase	0.832	0.837	0.872	**0.879**
Sonar	0.740	0.754	**0.778**	0.767
Average	0.760	0.795	0.808	**0.815**
Number of Wins	0	2	2	8

Table 3. Average values of G-mean for seven compared methods on multiple data sets shown in Table 1.

Dataset	MC	SM	SMB	ENN	CBOS	COG-OS	HPS-DRS
Glass	0.889	0.887	**0.892**	0.865	0.834	0.877	0.859
Letter	0.930	0.933	0.933	0.921	0.931	0.919	**0.967**
Vowel	0.986	0.972	0.979	0.967	0.971	0.982	**0.991**
Page	0.782	**0.809**	**0.809**	0.785	0.775	0.798	0.803
Satimage	0.813	0.829	0.835	0.821	0.776	0.809	**0.846**
Segment	0.971	0.976	0.982	0.968	0.975	0.944	**0.996**
Transfusion	0.581	0.583	0.594	0.581	0.568	**0.606**	0.567
Vehicle	0.625	0.688	0.683	0.635	0.669	0.633	**0.724**
German	0.618	0.641	**0.662**	0.589	0.605	0.643	0.656
Pima	0.649	0.669	0.688	0.631	0.639	0.609	**0.692**
Spambase	0.837	0.835	0.848	0.803	0.791	0.861	**0.879**
Sonar	0.728	0.731	0.745	0.711	0.764	0.755	**0.767**
Average	0.784	0.796	0.804	0.773	0.775	0.786	**0.812**
Number of Wins	0	1	3	0	0	1	8

Table 4. Average values of AUC for seven compared methods on multiple data sets shown in Table 1.

Dataset	MC	SM	SMB	ENN	CBOS	COG-OS	HPS-DRS
Glass	**0.991**	0.979	0.983	0.966	0.972	0.981	0.974
Letter	0.989	**0.999**	0.998	0.979	0.981	0.983	0.998
Vowel	**0.998**	0.993	**0.998**	0.989	0.988	0.996	0.996
Page	0.835	0.839	0.842	0.821	0.817	0.877	**0.885**
Satimage	0.945	0.926	0.942	0.902	0.917	0.941	**0.982**
Segment	0.998	**0.999**	0.997	0.998	0.993	**0.999**	**0.999**
Transfusion	0.751	0.755	0.774	0.750	0.751	**0.786**	0.744
Vehicle	0.981	0.989	0.992	0.979	0.984	0.975	**0.997**
German	0.715	0.728	**0.754**	0.695	0.697	0.741	0.748
Pima	0.744	0.766	0.779	0.731	0.724	0.741	**0.816**
Spambase	0.967	0.972	0.976	0.964	0.911	0.985	**0.993**
Sonar	0.779	0.808	0.822	0.781	0.795	0.822	**0.843**
Average	0.891	0.866	0.905	0.880	0.876	0.902	**0.915**
Number of Wins	2	2	2	0	0	2	7

Table 5. Average values of G-mean for HPS-DRS and SMB with optimized by measure metric on multiple data sets shown in Table 1.

Dataset	SMB		$HPS-DRS_{GM}$		$HPS-DRS_{AUC}$	
	G-mean	R_{os}	G-mean	α	G-mean	α
Glass	**0.907**	450%	0.879	0.85	0.871	0.8
Letter	0.939	250%	**0.974**	0.75	0.967	0.7
Vowel	0.991	350%	**0.996**	0.8	0.991	0.85
Page	**0.817**	300%	0.809	0.75	0.809	0.75
Satimage	0.851	350%	0.867	0.8	**0.875**	0.85
Segment	0.987	300%	**0.997**	0.55	0.996	0.7
Transfusion	**0.594**	200%	0.574	0.75	0.574	0.75
Vehicle	0.691	150%	**0.741**	0.6	0.724	0.7
German	**0.667**	250%	0.664	0.6	0.664	0.6
Pima	0.691	150%	**0.707**	0.8	0.695	0.75
Spambase	0.849	200%	0.879	0.65	**0.884**	0.6
Sonar	0.755	100%	**0.768**	0.3	**0.768**	0.3

Table 6. Experimental results (AUC) of tuning the class label noise level

Noise level	MC	SM	SMB	ENN	CBOS	COG-OS	HPS-DRS
10%	0.697	0.705	0.742	0.695	0.684	0.721	**0.745**
20%	0.685	0.682	**0.734**	0.684	0.697	0.703	0.729
30%	0.662	0.658	0.715	0.695	0.658	0.682	**0.717**
40%	0.645	0.649	0.691	0.695	0.633	0.665	**0.702**
50%	0.621	0.628	0.668	0.695	0.617	0.647	**0.679**

Table 7. Experimental results (AUC) of tuning the attribute noise level

Noise level	MC	SM	SMB	ENN	CBOS	COG-OS	HPS-DRS
10%	0.702	0.711	**0.747**	0.695	0.683	0.741	0.741
20%	0.707	0.708	0.729	0.684	0.657	0.716	**0.733**
30%	0.685	0.689	0.722	0.681	0.653	0.708	**0.735**
40%	0.669	0.672	0.704	0.667	0.636	0.671	**0.720**
50%	0.655	0.651	0.676	0.644	0.612	0.652	**0.708**

Table 8. Experimental results of candidate nodule classification

Methods	Sen.	Spec.	G-mean	AUC
CS-SVM	0.839	0.862	0.850	0.839
Asy-Adaboost	0.828	0.943	0.883	0.866
TL-SVM	1	0.656	0.810	0.847
HPS-DRS	0.871	0.939	**0.904**	**0.911**