Applying Adaptive Over-sampling Technique Based on Data Density and Cost-Sensitive SVM to Imbalanced Learning

Senzhang Wang[1,2], Zhoujun Li[1,2,*], Wenhan Chao[1,2] and Qinghua Cao[3]

Abstract—**Resampling method is a popular and effective technique to imbalanced learning. However, most resampling methods ignore data density information and may lead to overfitting. A novel adaptive over-sampling technique based on data density (ASMOBD) is proposed in this paper. Compared with existing resampling algorithms, ASMOBD can adaptively synthesize different number of new samples around each minority sample according to its level of learning difficulty. Therefore, this method makes the decision region more specific and can eliminate noise. What's more, to avoid over generalization, two smoothing methods are proposed. Cost-Sensitive learning is also an effective technique to imbalanced learning. In this paper, ASMOBD and Cost-Sensitive SVM are combined. Experiments show that our methods perform better than various state-of-art approaches on 9 UCI datasets by using metrics of G-mean and area under the receiver operation curve (AUC).**

Keywords-over-sampling; Cost-sensitive SVM; imbalanced learning

I. INTRODUCTION

In practical application, many datasets are imbalanced, i.e., some classes have much more instances than others. Imbalanced learning is common in many situations like information filtering [1] and fraud detection [2]. Datasets imbalance must be taken into consideration in classifier designing, otherwise the classifier may tend to be overwhelmed by the majority class and to ignore the minority class.

Resampling technique is an effective approach to imbalance learning. Many resampling methods are used to reduce or eliminate the extent of datasets imbalance, such as over-sampling the minority class, under-sampling the majority class and the combination of both methods. But [3] showed that under-sampling can potentially remove certain important instances and lose some useful information, and over-sampling may lead to overfitting. Over-sampling methods also suffer from noise and outliers [4].

Support Vector Machine (SVM) has been widely used in many application areas of machine learning. However, regular SVM is no longer suitable to imbalance-class especially when the datasets are extremely imbalanced. An effective approach to improve the performance of SVM used in imbalanced datasets is to bias the classifier so that it pays more attention to minority instances. This can be done by setting different misclassifying penalty [5].

We proposed an over-sampling algorithm based on data density in previous work [6]. However, this algorithm sometimes leads to overfitting. In this paper, an adaptive over-sampling algorithm with two smoothing methods to avoid overfitting is proposed. Compared with other over-sampling algorithms and our previous work, this algorithm can synthesize samples more efficiently and eliminate the effects of noise. Contributions of this paper are as follows:

--This novel method can effectively eliminate the noise compared with most other sampling methods like RO and SMOTE. Noise is recognized and no new samples are synthesized around it.

--Different number new samples are synthesized around each minority sample according to its level of learning difficulty. This level is related to the sample density information. To calculate the sample density, core-distance and reachability-distance are used [7]. We will elaborate this idea in section IV.

--To avoid overfitting, two smoothing methods are proposed. One is using a sigmoid function to smooth the disparity of new samples synthesized around each minority sample. The other is using linear interpolation method to tradeoff between our algorithm and SMOTE algorithm. Experiments show that both methods are effective.

The rest of the paper is organized as follows: Section II reviews related works. Section III gives an overview of performance measures. Section IV details our approach. Section V presents experimental results comparing our approach with other approaches. Section VI discusses the result and concludes this paper.

II. RELATED WORK

Resampling techniques are widely used in imbalanced learning such as random over-sampling (RO), random undersampling (RU) and over-sampling with informed generation of new samples. [4] proposed an algorithm-SMOTE to over-sampling minority datasets. This algorithm synthesizes new samples along the line between the minority and their selected nearest neighbors. The disadvantage of SMOTE is that it makes the decision regions larger and less specific [8]. SMOTE-ENN and SMOTE-Tomek are two popular methods combining sampling technique and data cleaning technique. Experiments in [3] show that these two

[1]State Key Laboratory of Software Development Environment, Beihang University, Beijing, China.
[2]Beijing Key Laboratory of Network Technology, Beihang University, Beijing, China.
[3]Department of Computer Science and Engineering, Beihang University, Beijing, China.
*Corresponding author: lizj@buaa.edu.cn

methods perform better than SMOTE, especially for the highly imbalanced datasets. Many other informed sampling method are proposed in [9][10][11]. EasyEnsemble and BalanceCascade algorithms in [9] are two effective informed under-sampling methods.

Cost-sensitive learning is also an effective solution. This algorithm can improve the performance of classification by setting different misclassification cost to the majority and minority datasets. [5] suggested using different penalty constants for different classes of data in SVM. Decision tree is also used to imbalanced learning. [12] identified a new skew Hellinger distance in class imbalance. A new solution based on cost-sensitive SVM is proposed in [13]. Additional margin compensation is further included to achieve a more accurate solution in this method.

Many algorithms combining resampling and cost-sensitive learning have also been proposed. [14] combined SOMTE algorithm with cost-sensitive SVM. SMOTEBoost combined SMOTE algorithm with cost-sensitive boosting algorithm for class-imbalance learning [15]. A SVMs modeling method for highly imbalanced classification is proposed in [16]. The modeling method incorporates different "rebalance" heuristics in SVM modeling, including cost-sensitive learning, and over- and under-sampling. A fuzzy support vector machine (FSVM) is proposed in [17] for imbalance learning. FSVM can handle the problem of outliers and noise. For more information of imbalanced learning, [18] is a good survey.

Compared with most other resampling methods ignoring datasets density information, we propose a novel adaptive over-sampling algorithm based on datasets density (ASMBOD). To avoid overfitting, two smoothing methods are also proposed. Experiments on 9 UCI datasets show that ASMOBD performs better than various state-of-art approaches by metrics of G-mean and area under the receiver operation curve (AUC). Experiments also show that the combination of ASMOBD and Cost-Sensitive SVM can further improve the performance of imbalanced learning.

III. PERFORMANCE MEASURE

For highly imbalanced datasets, accuracy is no longer a reasonable metric. For example, datasets with imbalance rate of 99 (the ratio between majority sample number and minority sample number), a classifier that classifies all the instances negative will be 99% accurate, but it will be completely useless as a classifier. The performance of the machine learning algorithm is typically evaluated by a confusion matrix as illustrated in Table I.

TABLE I. CONFUSION MATRIX

	Predicted Positive Class	**Predicted Negative Class**
Actual Positive Class	TP (True Positive)	FN (False Negative)
Actual Negative Class	FP (False Positive)	TN (True Negative)

[18] suggested the G-mean metric to evaluate the performance of classifier, which is often used by researchers. AUC has been proved to be a reliable performance measure for imbalanced and cost-sensitive learning [20].

In this paper, we use G-mean and AUC as performance measures. The metrics and some other parameters we used are defined as follows:

$$acc_+ = sensitivity = \mathrm{Re}call = TP/(TP+FN) \quad (1)$$

$$acc_- = specificity = TN/(TN+FP) \quad (2)$$

$$G-mean = \sqrt{acc_+ \cdot acc_-} = \sqrt{sensitivity \cdot specificity} \quad (3)$$

ROC curves can be thought of as the representative of the family of best decision boundaries for relative costs of TP and FP. On the ROC curve the X-axis represents FPR = FP/ (TN + FP) and the y-axis represents TPR = TP/ (TP + FN). AUC is the area below the curve. Figure 1 shows an illustration. Figure 1 is the ROC of *sick* dataset in UCI repository. The line y=x represents the scenario of random guess. The larger the AUC is the better the performance of classifier is.

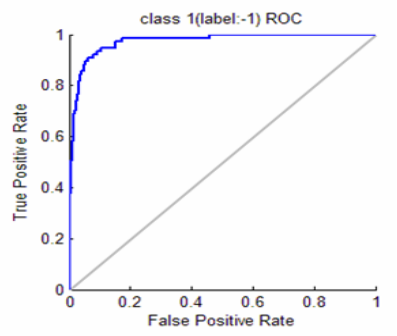

Figure 1: Example of ROC curve. The line y=x represents randomly guessing. The more the ROC tilts toward the top-left corner, the larger the AUC of the ROC and the better performance of the classifier.

IV. ASMOBD AND ASMOBD-CS

Regular over-sampling algorithms, like over-sampling with random replacement and SMOTE, ignore the density and distribution information of the datasets and suffer from the problem of outliers and noise.

SMOTE-ENN and SMOTE-Tomek are two effective algorithms to eliminate the noise in comparison to SMOTE, but experiments in [21] show that these algorithms may not provide better performance than random over-sampling when the number of minority class is large. In this section, firstly, the novel over-sampling algorithm ASMOBD we proposed will be described in detail; secondly, to avoid overfitting, two smoothing methods are proposed; thirdly, SVM with different error costs will be described briefly.

A. Adaptive Over-sampling Technique Based on samples Density (ASMOBD)

The proposed algorithm can adaptively synthesize different number new samples around each minority sample according to its level of learning difficulty. As shown in Fig. 2, the minority sample's level of learning difficulty depends on both the local minority sample density and local majority sample density. What's more, to avoid over generalization, an imbalanced factor which is determined by the ratio

between local majority and minority samples is proposed.

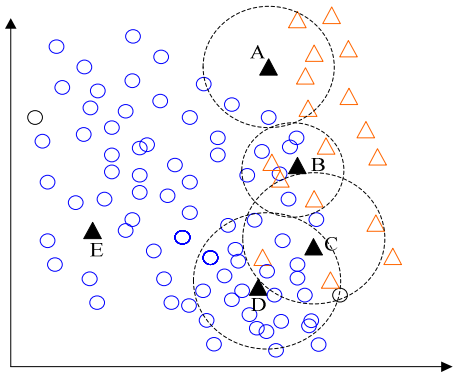

Figure 2: Example of ASMOBD. Triangles and circles represent samples of two different classes. Dash circles represent core-distance of sample A, B, C, D and E. A is easy to be classified correctly, so a small number of new samples will be synthesized. B and C are hard to be classified, so more new samples will be synthesized around them. Few new samples will be synthesized around D in order to avoid overfitting, because too many majority samples are in its core-distance. E is noise, so no new samples will synthesized around it.

Fig. 2 shows the intuitive idea of our method. The triangles represent the minority samples, the circles represent the majority samples and the black triangles represent the minority samples around which new samples will be synthesized (except noise E). The level of learning difficulty for minority samples is determined by three factors: the local density of minority samples, the local density of majority samples and the local imbalanced ratio. The minority sample is easy to be classified when its local minority density is large and local majority density is small. On the contrary, the minority sample is hard to be classified when its local minority density is small and local majority density is large.

More new samples will be synthesized around the minority samples which are harder to be classified correctly while less new samples will be synthesized around the minority samples which are easier to be classified correctly. To avoid over generalization, an imbalanced factor which is determined by the ratio between local majority and minority samples is proposed. In Fig. 2, though sample D is very hard to be classified correctly, less new samples will be synthesized around it for its high imbalanced ratio. For noise, no new samples will be synthesized around it like sample E.

A data density based clustering algorithm OPTICS was proposed in [7], which could identify clustering structure. The core-distance and reachability-distance proposed in this algorithm fully reflect the density information of the datasets and noisy samples will be judged by the two distance. The core-distance and reachability-distance are defined as follows:

Definition 1 [7]. *core-distance* of an object p: Let p be an object from a database D, let ε be a distance value, let $N_\varepsilon(p)$ be $\varepsilon-neighborhood$ of p, let $card(N_\varepsilon(p))$ denote the cardinality of the set $N_\varepsilon(p)$, let $MinPts$ be a natural number and let $MinPts-distance(p)$ be the distance from P to its $MinPts$' neighbor. Then, the *core-distance* of p is defined as $core-distance_{\varepsilon,MinPts}(p) =$

$$\begin{cases} UNDEFINED & \text{if } Card(N_\varepsilon(p)) < MinPts \\ MinPts-distance(p) & \text{else} \end{cases}$$

Definition 2 [7]. *reachability-distance* object p w.r.t. object o: Let p and o be objects from a database D, let $N_\varepsilon(o)$ be the $\varepsilon-neighborhood$ of o, and let $MinPts$ be a natural number. Then, the *reachability-distance* of p with respect to o is defined as $reachability-distance_{\varepsilon,MinPts}(p,o) =$

$$\begin{cases} UNDEFINED & \text{if } N_\varepsilon(o) < MinPts \\ max(core-distance(o), distance(o,p)) & \text{else} \end{cases}$$

Intuitively, the core-distance of an object p is the smallest distance between p and an object in its $\varepsilon-neighborhood$ such that p would be a core object. The reachability-distance of an object p with respect to another object o is the smallest distance such that p is directly density-reachable from o if o is a core object.

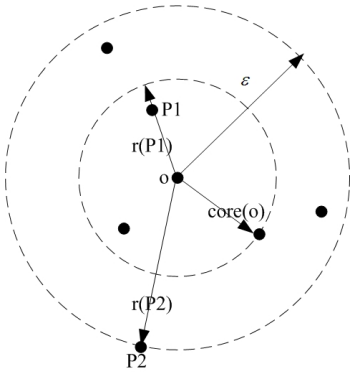

Figure 3: Core-distance(o), reachability-distance r(P1,o), r(P2,o) for MinPts=4

Fig.3 shows an example of core-distance and reachability-distance of an object o. Core-distance and reachability-distance of an object fully reflect the density information of each object. Noise can also be defined according to the two distances.

Definition 3. Noise based on *core-distance* and *reachabiliy-distance*: Let p and o be object from a database D. If $core-distance_{\varepsilon,MinPts}(p)$ is beyond the threshold T_1, and $reachability-distance_{\varepsilon,MinPts}(p,o)$ is beyond the threshold T_2 (T_1 and T_2 are all predefined), then p is a noise.

$$NOISE_p = \begin{cases} 1 & \text{if } core-distance_{\varepsilon.MinPts}(p) > T_1 \text{ and} \\ & reachability-distance_{\varepsilon,o}(p,o) > T_2 \\ 0 & \text{else} \end{cases}$$

According to the above definitions, Density of new samples synthesized around each minority sample is computed by the formula below:

$$DF_i = \eta * \varepsilon_i + (1-\eta) * card_{majority}(N_{\varepsilon_i}(x_i)). \quad (4)$$

DF_i means the density of new samples synthesized around the sample x_i. ε_i represents the core-distance of sample x_i. $card_{majority}(N_{\varepsilon_i}(x_i))$ is the number of majority samples which are included in the hyper sphere with a radius of core-distance of sample x_i. ε_i and $card_{majority}(N_{\varepsilon_i}(x_i))$ are all normalized. η is weighting coefficient tradeoff

between ε_i and $N_{i\varepsilon}$. We set η to 0.5, and in most cases, 0.5 is an appropriate value for η in our experiment.

The number of new samples synthesize around each samples is computed by the formula below:

$$N_i = \frac{DF_i * N}{\sum_{j=1}^{n} DF_i} \tag{5}$$

N_i means the number of new samples synthesized around sample x_i. N means the total number of new samples synthesized. The algorithm of synthesizing new samples is similar to SMOTE. The only difference is no new samples are synthesized around noisy samples and the number of new samples around each minority sample is different. The noisy sample is judged by the formula below:

$$NOISE_i = \begin{cases} 1 \text{ if } CD_i > \frac{1}{N\_min}\sum_{i=1}^{N\_\min} CD_i * t_1 \text{ and} \\ \quad RD_i > \frac{1}{N\_min}\sum_{i=1}^{N\_\min} RD_i * t_2 \\ 0 \text{ else} \end{cases} \tag{6}$$

$NOISE_i$ is 1 if x_i is a noisy sample and 0 if not. Array CD[] and RD[] store the core-distance and reachability-distance of all the minority samples. N_min is the number of minority samples. t_1 and t_2 are noise threshold coefficients predefined. In our experiment, we set t_1 and t_2 to 4. It means than if the core-distance and reachability-distance of sample x_i is 4 times larger than the average core-distance and reachability-distance, sample x_i is considered to be a noise.

B. Sigmoid Function Smoothing

The method we proposed above can effectively synthesize different number of new samples around each minority sample according to its level of learning difficulty. However, overfitting may still exist. For example, some minority samples will synthesize a large number of new samples while some minority samples will not synthesize any new samples. To address this problem, two smoothing methods are proposed.

The first approach is using a sigmoid function to smoothing equation (4). Smoothing details is as follows:

$$\begin{aligned} DF\_S_i &= DF_i - Balance\_ratio_i \\ &= \eta * \varepsilon_i + (1-\eta) * card_{majority}(N_{\varepsilon_i}(x_i)) \\ &\quad - Balance\_ratio_i \end{aligned} \tag{7}$$

$$Balance\_ratio_i = abs\left(\frac{2}{1+\exp\{-a * Ratio_i\}} - 1\right) \tag{8}$$

$$Ratio_i = \frac{Maj\_num_i}{Min\_num_i} \tag{9}$$

DF_i is smoothed by subtracting a value $Balance\_ratio_i$, which is calculated by a sigmoid function. In this function, a is a weighting coefficient and $Ratio_i$ is the imbalance ratio between the local majority samples and local minority samples. In our experiment, a is set to 0.05 or 0.1. Maj_num_i and Min_num_i mean the number of majority samples and minority samples in the $\varepsilon - neighbourhood$ of sample x_i respectively.

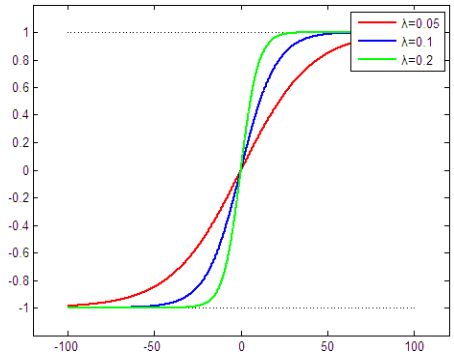

Figure 4: An example of sigmoid function

Fig.4 shows an example of sigmoid function curves with parameter $\lambda = 0.05$, $\lambda = 0.1$ and $\lambda = 0.2$ respectively. The curves show that the larger λ is, the steeper the curve is. The function value tends to ± 1 when the argument tend to positive infinity and negative infinity. In equation (8), a is the value of parameter λ. The larger the value of $Ratio$ is, the more penalty will exert to DF_i. Intuitively, if the sample imbalanced ratio is too large, the sigmoid function will generate more penalties to avoid too many new samples synthesized around sample x_i.

C. Linear Interpolation Smoothing

The second smoothing approach we proposed is linear interpolation. This approach is a combination between SMOTE and ASMOBD. SMOTE ignores the data density information and ASMOBD may lead to overfitting because of overemphasizing data density information. We combine both methods.

The linear interpolation smoothing is proposed as follows:

$$DF\_L_i = \mu * DF_i + (1-\mu) * k \tag{10}$$

In equation (10), μ is the smoothing coefficient. k is the proportion of over-sampling with SMOTE method. Equation (10) shows that linear interpolation smoothing method is a compromise between SMOBD and SMOTE. Intuitively, this method is a compromise between individuation and generality.

Pseudo code of our algorithm is as follows:

1. **procedure AS_S/AS_LI**
2. *Input: Dataset D*
3. *Output: new Dataset* D_{new}
4. *//Calculate the core-distance and reachability-distance of each minority sample using OPTICS*

$$\begin{pmatrix} core-distance(D_{minority}) \\ reachability-distance(D_{minority}) \end{pmatrix} \leftarrow OPTICS(D,k,threshold)$$

5. Eliminate noise according to *core-distance* and *reachability-distance* using Definition 3 and formula (6).
6. *//Calculate the new synthesized samples density of each minority sample* x_i

$$DF_i = \eta * \varepsilon_i + (1-\eta) * card_{majority}(N_{\varepsilon_i}(x_i)).$$

7. Smoothing

   Method 1: $DF\_S_i = DF_i - Balance\_ratio_i$

   Method 2: $DF\_L_i = \mu * DF_i + (1-\mu) * k$

8. *//Calculate the number of new synthesized samples around each minority sample* x_i

$$N_i = \frac{DF_i * N}{\sum_{j=1}^{n} DF_i}$$

9. Synthesize new samples similar to SMOTE method, and output new dataset D_{new}
10. **end procedure**

Fig.5 shows an example of the results of the two smoothing methods we proposed. We use the *sick* dataset of UCI in this example. k value is 400%. The horizontal axis represents each minority sample and the vertical axis represents the number of new samples synthesized around each minority sample.

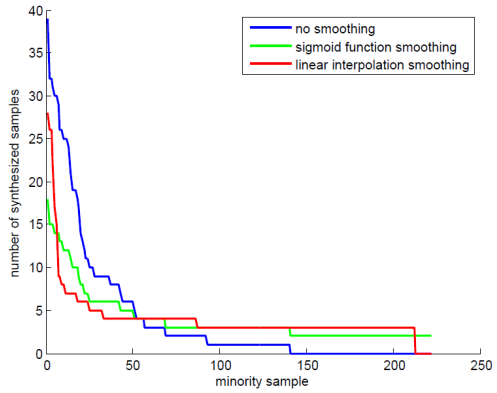

Figure 5: An example of the results of two smoothing methods

This example demonstrates without smoothing, large number of new samples will be synthesized around a few minority samples while no new samples will be synthesized around some minority samples. The two smoothing methods effectively reduce the imbalance.

For the linear interpolation smoothing method, a difficult, yet important problem is how to determine the value of parameter μ. Some experiments are made to test how parameter μ influences the performance of classification.

Three datasets of UCI are used here: *hypothyroid*, *abalone* and *sick*. We use the G-mean value as performance measure. The experiments results are as follows:

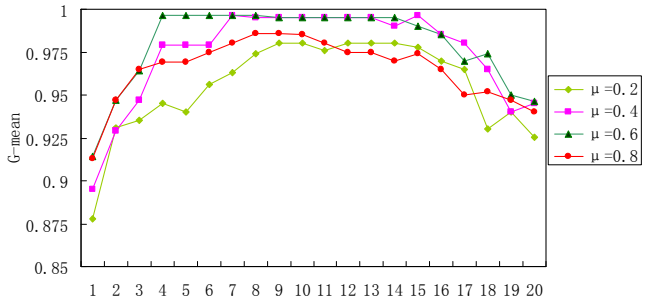

Figure 6: G-mean for diffenrent μ values for hypothyroid dataset

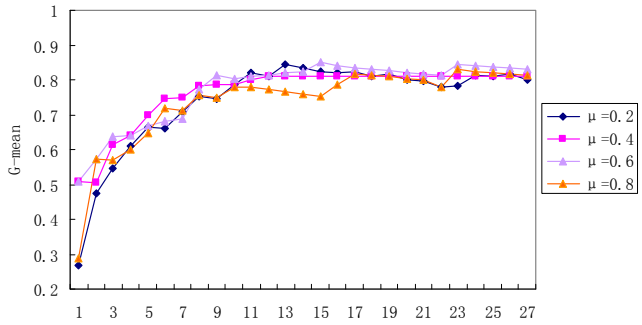

Figure 7: G-mean for diffenrent μ values for abalone dataset

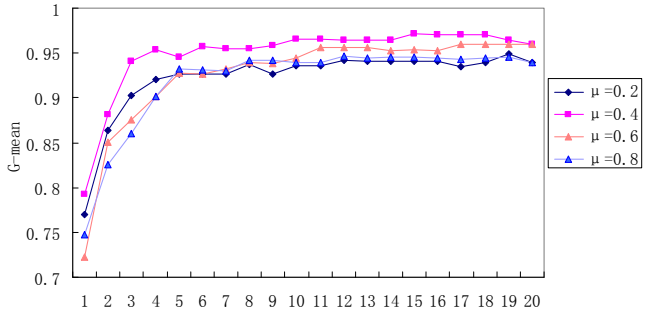

Figure 8: G-mean for diffenrent μ values for sick dataset

The horizontal axis shows the proportion of over-sampling and the vertical axis shows the G-mean value of the method we proposed with linear interpolation smoothing. Four different μ values are tested in our experiment, μ =0.2, μ =0.4, μ =0.6 and μ =0.8. The experiments results show that μ =0.4 and μ =0.6 are better than μ =0.2 and μ =0.8. For μ =0.2, SMOTE overwhelms ASMOBD and for μ =0.8, ASMOBD overwhelms SMOTE. The experiments results are also in accordance with intuition: the combination of two methods can improve performance.

D. SVM with different error costs and ASMOBD-CS

As mentioned above, regular SVM is invalid to the imbalanced datasets. [14] showed that with imbalanced datasets, the learned boundary is too close to the minority samples, so SVM should be biased in a way that will push the boundary away from the positive samples. [5] suggested using different error costs for the positive (S_+) and negative (S_-) classes. The classifier function equates to solve the quadratic programming problem as follows:

$$\min \quad J(\omega,\omega_0,\xi)=\frac{1}{2}\left\|\omega^2\right\|+C^+\sum_{\{i|y_i=+1\}}^{n_+}\xi_i+C^-\sum_{\{i|y_i=-1\}}^{n_-}\xi_i \qquad (11)$$

$$\text{Subject to:} \quad y_i(\omega^T x_i+\omega_0)\geq 1-\xi_i \qquad (12)$$

$$\xi_i\geq 0,\ i=1,\ldots,n_-+n_+ \qquad (13)$$

In (11), C^+ represents the cost of misclassifying the positive sample and C^- represents the cost of misclassifying the negative sample. It was reported in [9] that the optimal result could be obtained when C^-/C^+ equals to the minority-to-majority class ratio. The C^-/C^+ in our method is determined by formula below:

$$\frac{C^-}{C^+}=\frac{Num\_Minority*k}{Num\_Majority} \qquad (14)$$

$Num\_Minority$ is the number of minority samples, k is the proportion of over-sampling and $Num\_Majority$ is the number of majority samples.

With different error costs, the boundary is pushed more towards the majority samples. [14] showed that SVM with different error costs may obtain stronger cues from the majority samples than from the minority samples about the orientation of the plane. Consequently, the combination of the two methods can achieve better performance.

ASMOBD-CS combines ASMOBD and cost-sensitive SVM. Though imbalance rate is reduced by over-sampling, it still exists. To further reduce the imbalance rate, different error costs are proposed according to the reduced imbalance rate.

V. Experiment and Result

A. Datasets

9 UCI datasets are used to test the algorithms we proposed. Information about these datasets is summarized in Table II. When more than two classes exist in the dataset, one class is considered to be positive and all the other classes are considered to be negative.

In our experiments, G-mean and area under curve (AUC) are used as metrics. For each dataset, we perform 5-fold cross validation. In each fold four out of five samples are selected to be training set, and the left one out of five samples is testing set. This process repeats 5 times so that all samples are selected in both training set and testing set.

TABLE II. TEST DATASETS INFORMATION TABLE

Dataset	#Attributes	#Positive	#Negative	#Imbalance Ratio
abalone	9	32	4145	130
hypothyroid	29	95	3677	39
sick	29	231	3541	15
glass	9	29	185	6
car	6	69	1659	24
prima	8	268	500	2
hepatitis	19	32	123	4
segment	19	330	1980	6
auto-mgp	7	68	324	5

B. Experiment between ASMOBD and other over-sampling methods

We compare our methods with SMOTE and random over-sampling (RO). In our experiments, we use ASMOBD with two smoothing methods. The sigmoid function smoothing method is noted as AS_S and the linear interpolation smoothing method is noted as AS_LI.

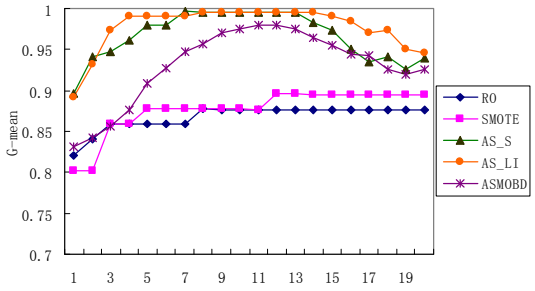

Figure 9: G-mean for hypothyroid dataset

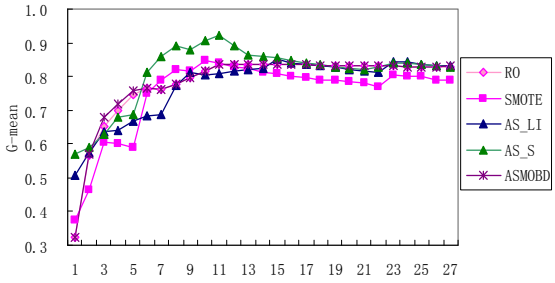

Figure 10: G-mean for abalone dataset

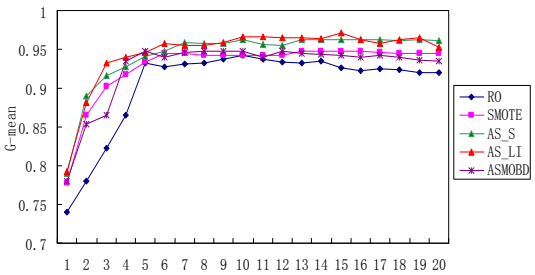

Figure 11: G-mean for sick dataset

Fig 9, Fig 10 and Fig 11 show the G-mean metric of each method for *hypothyroid*, *abalone* and *sick* datasets of UCI repository. The experiments results show that the AS_S and AS_LI methods we proposed are both better than RO and SMOTE methods for almost all over-sampling proportions for the three datasets we used.

C. *Experiment between ASMOBD and ASMOBD with Cost-sensitive SVM*

We also make experiments between ASMOBD and ASMOBD with cost-sensitive SVM. These experiments indicate the combination of over-sampling method and cost-sensitive learning can further improve the performance of imbalanced learning.

The two methods we proposed above with cost-sensitive SVM are noted as AS_S_CS and AS_LI_CS respectively. *hypothyroid* and *abalone* datasets are used in the experiments. The SMOTE with cost-sensitive SVM method (SDC) [14] is used to be a baseline method.

Fig 12 and Fig 13 show the G-mean metric of the experiment. The results indicate that the combination of over-sampling and cost-sensitive learning can further improve the G-mean metric of imbalanced learning. However, the improvement is not remarkable in our experiments. Moreover, with the increasing of over-sampling proportion, the performance of the two methods is not that big of a difference. Experiment also shows that AS_S_CS and AS_LI_CS are both better than SDC in the G-mean metric in the two datasets.

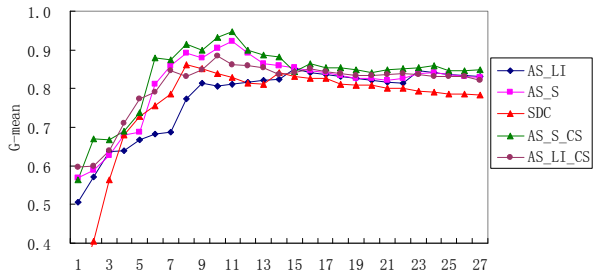

Figure 12: G-mean for abalone dataset

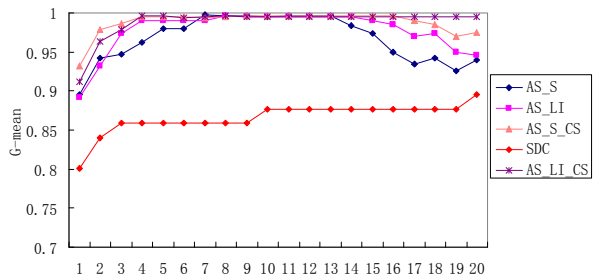

Figure 13: G-mean for hypothyroid dataset

D. *Experiment among 9 methods using 9 datasets*

We compared 9 methods: Random over-sampling (RO), SMOTE, Borderline SMOTE, cost-sensitive SVM, SMOTE with cost-sensitive SVM (SDC), AS_S, AS_LI, AS_S_CS and AS_LI_CS, on 9 datasets described in TABLE II.

For each dataset, we calculate the G-mean and AUC metrics under different over-sampling proportions and average them. For example, the over-sampling proportion is from 100% to 1500% for *abalone* dataset. We sample from 100% to 1500% and calculate the average G-mean and AUC. The upper limit of over-sampling proportion depends on the imbalance ratio of each dataset.

G-mean and AUC metrics described in section III are both used in our experiments. The best experiments results are denoted by bold body and black underlines. Experiments results are shown in Table III and Table IV. Table III is the G-mean metric table and Table IV is the AUC metric table.

Results demonstrate that in eight out of nine datasets, AS_S_CS and AS_LI_CS have the highest G-means and AUC value. Comparison among RO, SMOTE, Borderline SMOTE, AS_S and AS_LI demonstrates that the two methods we proposed outperform other methods in most datasets. Experiments demonstrate AS_S and AS_LI synthesize new samples more effectively than SMOTE. SMOTE and other algorithms synthesize the same number of new samples around each minority sample, so some useful information is lost. Moreover, noise is an important factor to influence the performance improvement for SMOTE. The comparison between AS_S, AS_LI and AS_S_CS, AS_LI_CS demonstrates the combination of over-sampling and cost-sensitive SVM can further improve the performance of classifier.

VI. CONCLUSION

A novel adaptive over-sampling technique based on data density information is proposed in this paper. We also combine the new over-sampling method with cost-sensitive SVM. Empirical results show that our methods perform better than state-of-art approaches like RO, SMOTE, Borderline SMOTE and SDC on a variety of datasets by using G-mean, area under the receiver operation curve (AUC) metrics.

Though ASMOBD and ASMOBD-CS can achieve better performance in most cases, many problems still need to be addressed. Firstly, there are some parameters in our algorithms. The performance of our algorithms varies a lot with different values of parameters. How to find the best parameter value to achieve the best performance is a problem we need to solve in the future work. Secondly, more time is needed to synthesize the same number of new samples in our method than SMOTE. To compute density of each sample, reachability-distance and core-distance need to be computed firstly, which consumes much more time than that of SMOTE. Computation complexity reduction is another work we need to do in the future.

ACKNOWLWDGEMENT

This work was supported by the National Natural Science Foundation of China [grant number 60973105, 90718017, 61170189], the Research Fund for the Doctoral Program of Higher Education [grant number 20111102130003] and the Fund of the State Key Laboratory of Software Development Environment [grant number SKLSDE-2011ZX-03].

TABLE III. G-MEAN METRIC OF EXPERIMENT RESULT

Dataset	Over-sampling proportion	RO	SMOTE	Bordeline SMOTE	CS	AS_S	AS_LI	SDC	AS_S_CS	AS_LI_CS
abalone	100% - 1500%	0.834	0.802	0.840	0.340	0.851	0.826	0.817	**0.869**	0.844
hypothyroid	100% - 1000%	0.860	0.860	0.920	0.781	0.969	0.974	0.863	**0.986**	0.982
sick	100% - 1000%	0.881	0.912	0.910	0.671	0.925	0.928	0.925	0.938	**0.940**
glass	100% - 500%	0.910	**1.000**	0.992	0.992	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
car	100% - 1000%	0.977	0.966	0.985	0.460	0.980	0.981	0.981	0.985	**0.986**
prima	100% - 500%	0.702	0.700	0.710	0.673	0.709	0.711	0.702	0.718	**0.721**
hepatitis	100% - 500%	0.873	0.869	**0.940**	0.542	0.900	0.894	0.890	0.913	0.910
segment	100% - 1000%	0.910	0.917	0.965	0.842	0.955	0.960	0.932	0.974	**0.982**
auto-mgp	100% - 800%	0.890	0.924	0.942	0.575	0.940	0.934	0.930	0.952	**0.963**

TABLE IV. AUC METRIC OF EXPERIMENT RESULT

Dataset	Over-sampling proportion	RO	SMOTE	Bordeline SMOTE	CS	AS_S	AS_LI	SDC	AS_S_CS	AS_LI_CS
abalone	100% - 1500%	0.828	0.837	0.842	0.67	0.854	0.823	0.831	**0.864**	0.842
hypothyroid	100% - 1000%	0.869	0.869	0.927	0.804	0.971	0.980	0.863	**0.982**	0.973
sick	100% - 1000%	0.888	0.914	0.915	0.723	0.927	0.923	0.924	**0.937**	0.936
glass	100% - 500%	0.924	**1.000**	0.990	0.990	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
car	100% - 1000%	0.980	0.967	0.985	0.600	0.981	0.984	0.982	0.986	**0.987**
prima	100% - 500%	0.738	0.735	0.720	0.698	0.740	0.745	0.737	0.742	**0.747**
hepatitis	100% - 500%	0.875	0.870	**0.944**	0.607	0.900	0.894	0.893	0.918	0.910
segment	100% - 1000%	0.917	0.922	0.960	0.874	0.960	0.962	0.943	**0.980**	**0.980**
auto-mgp	100% - 800%	0.900	0.930	0.945	0.662	0.944	0.940	0.932	0.954	**0.964**

REFERENCES

[1] D. Lewis, J. Catlett , "Training Text Classifiers by Uncertainty Sampling," In Proceedings of 17th International ACM SIGIR Conference(1994)

[2] T. Fawcett, F. Provost, "Adaptive fraud detection," Data Mining and Knowledge Disvovery 1, 291-361(1997)

[3] G. E. A. P. A. Batista, R. C. Prati, M. C. Monard, "A study of the Behavior of several methods for balancing machine learning data," SIGKDD Explorations, 6(1):20-29(2004)

[4] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, "SMOTE:Synthetic Minority Oversampling Technique," Journal of Artificial Intelligence Research, pp, 321-357(2002)

[5] K. Veropoulos, C. Campbell, N. Cristianini, "Controlling the sensitivity of support vector machines," Proceedings of the International Joint Conference on AI(1999)

[6] Q. H. Cao, S. Z. Wang, "Applying Over-sampling Technique Based on Data Density and Cost-sensitive SVM to Imbalanced Learning," Proceedings of the 2011 International Conference on Information Management, Innovation Management and Industrial Engineering (2011).

[7] M. Ankerst, M. Breuning, H. P. Kriegel, J. Sander, "OPTICS: Ordering points to identity clustering structure," In Proceedings of the ACM SIGMOD Conference, pp. 49-60(1999)

[8] B. Chumphol, S. Krung, L. Chidchanok, "Safe-Level-SMOTE," Advances in Knowledge Discovery and Data Mining (PAKDD), pp. 475-482(2009)

[9] X. Y. Liu, J. X. Wu, Z. H. Zhou. "Exploratory Undersampling for Class-Imbalance Learning," In Proceedings of the 6th International Conference on Data Mining, pp.965-969(2006)

[10] H. Han, W. Y. Wang, B. H. Ma, "Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning," Proc. Int'l Conf. Intelligent Computing, pp. 878-887(2005)

[11] H. Haibo, B. Yang, A. G. Edwardo, L. Shutao, "ADASYN: Synthetic Sampling Approach for Imbalanced Learning," Int'l J. Conf. Neural Networks, pp. 1322-1328(2008)

[12] D. Cieslak, N. Chawla, "Learning Decision Trees for Unbalanced Data," In: Proceedings of ECML PKDD 2008 Part I. 241-256(2008)

[13] C. Y. Yang, J. Wang, J. S. Yang, "Imbalanced SVM Learning with Margin Compensation," In: Proceeding of Advances in Neural Networks 2008 (ISNN). 636-644(2008)

[14] R. Akbani, S. Kwek, N. Japkowicz, "Applying Support Vector Machines to Imbalanced Dtasets," In Proceedings of the 15th European Conference on Machine Learning, Italy, pp. 39-50(2004)

[15] V. C. Nitesh, L. Aleksandar, O. H. Lawrence, W. B. Kevin, "SMOTEBoost: Improving prediction of the minority class in boosting," In Proceedings of the 7th European Conference on Principles and Knowledge Discovery Databases, CavtatDubrovnik, Croatia, pp. 107-119(2003)

[16] Y. Tang, Y.Q. Zhang, N. V. Chawla, S. Krasser, "SVMs Modeling for Highly Imbalanced Classification," IEEE Trans. Sys, Man, Cyb. Part B(2009)

[17] B. Batuwita, V. Palade,: "FSVM-CLI: Fuzzy Support Vector Machines for Class Imbalance Learning ," IEEE Trans. Fuz. Sys, vol. 18, no. 3, Sep(2009)

[18] H. Haibo, A. G. Edwardo, "Learning from imbalanced data," IEEE Trans. Knowl. Data Eng., vol. 21, no. 9, pp. 1263-1284, Sep(2009)

[19] M. Kubat, S. Matwin, "Addressing the Curse of Imbalanced Training Sets: One-Sided Selection" Proceedings of the 14th International Conference of Machine Learning.

[20] T. Fawcett, "ROC graphs: Notes and practical considerations for researchers," HP Labs, Palo Alto, CA, Tech. Rep. HPL-2003-4,203

[21] J. Luengo, A. Fernandez, S. Garcia, "Addressing data complexity for imbalanced data sets: analysis of SMOTE-based oversampling and evolutionary undersampling," Soft Computing, 2010