SVMs Modeling for Highly Imbalanced Classification

Yuchun Tang, *Member, IEEE,* Yan-Qing Zhang, *Member, IEEE,* Nitesh V. Chawla, *Member, IEEE,* and Sven Krasser, *Member, IEEE*

***Abstract*— Traditional classification algorithms can be limited in their performance on highly unbalanced datasets. A popular stream of work for countering the problem of class imbalance has been application of a sundry of sampling strategies. In this work, we focus on designing modifications to SVM to appropriately tackle the problem of class imbalance. We incorporate different "rebalance" heuristics in SVM modeling including cost-sensitive learning, oversampling, and undersampling. These SVM based strategies are compared with various state-of-the-art approaches on a variety of datasets by using various metrics, including G-mean, Area Under ROC Curve (AUC-ROC), F-measure, and Area Under Precision/Recall Curve (AUC-PR). We show that we are able to surpass or match the previously known best algorithms on each dataset. In particular, of the four SVM variations considered in this paper, the novel *Granular Support Vector Machines - Repetitive Undersampling* algorithm (GSVM-RU) is the best in terms of both effectiveness and efficiency. GSVM-RU is effective as it can minimize the negative effect of information loss while maximizing the positive effect of data cleaning in the undersampling process. GSVM-RU is efficient by extracting much less support vectors, and hence greatly speeding up SVM prediction.**

***Index Terms*— highly imbalanced classification, cost-sensitive learning, oversampling, undersampling, computational intelligence, support vector machines, granular computing.**

I. Introduction

Mining highly unbalanced datasets, particularly in a cost-sensitive environment, is among the leading challenges for knowledge discovery and data mining [1], [2]. The class imbalance problem arises when the class of interest is relatively rare as compared to the other class(es). Without loss of generality we will assume that the positive class (or class of interest) is the minority class, and the negative class is the majority class. Various applications demonstrate this characteristic of high class imbalance, such as bioinformatics, e-business, information security, to national security. For example, in the medical domain the disease may be rarer than normal cases; in business the defaults may be rarer than good customers, etc. For our work on the Secure Computing TrustedSource network reputation system (http://www.trustedsource.org), we have to address the high imbalance towards malicious IP addresses. In addition, rapid classification is paramount as most malicious machines are only active for a brief period of time [3].

Manuscript received August 09, 2007; revised February 20, 2008; accepted July 09, 2008.

Yuchun Tang and Sven Krasser are with Secure Computing Corporation, 4800 North Point Parkway, Suite 300, Alpharetta, GA 30022. Yan-Qing Zhang is with Dept. of Computer Science, Georgia State University, Atlanta, GA 30302-3994. Nitesh Chawla is with Dept. of Computer Science & Engg., University of Notre Dame, IN 46556.

Sampling strategies, such as oversampling and undersampling, are extremely popular in tackling the problem of class imbalance. That is, either the minority class is oversampled or majority class is undersampled or some combination of the two is deployed. In this paper, we focus on learning Support Vector Machines (SVM) with different sampling techniques. We focus on comparing the methodologies on the aspects of *effectiveness* and *efficiency*. While, effectiveness and efficiency can be application dependent, in this work we define them as follows:

Definition 1: Effectiveness means the ability of a model to accurately classify unknown samples, in terms of some metric.

Definition 2: Efficiency means the speed to use a model to classify unknown samples.

SVM embodies the structural risk minimization principle to minimize an upper bound on the expected risk [4], [5]. Because structural risk is a reasonable trade-off between the training error and the modeling complication, SVM has a superior generalization capability. Geometrically, the SVM modeling algorithm works by constructing a separating hyperplane with the maximal margin. Compared with other standard classifiers, SVM is more accurate on moderately imbalanced data. The reason is that only Support Vectors (SVs) are used for classification and many majority samples far from the decision boundary can be removed without affecting classification [6]. However, an SVM classifier can be sensitive to high class imbalance, resulting in a drop in the classification performance on the positive class. It is prone to generating a classifier that has a strong estimation bias towards the majority class, resulting in a large number of false negatives [6], [7].

There have been some recent works in improving the classification performance of SVM on unbalanced datasets [8], [6], [7]. However, they do not address efficiency very well, and depending on the strategy for countering imbalance, they can take a longer time for classification than a standard SVM. Also, SVM can be slow for classification on large datasets [9], [10], [11]. The speed of SVM classification depends on the number of SVs. For a new sample X, $K(X, SV)$, the similarity between X and SV, is calculated for each SV. Then it is classified using the sum of these kernel values and a bias. To speed up SVM classification, one method is to decrease the number of SVs.

We previously presented a preliminary version of the *Granular Support Vector Machines - Repetitive Undersampling* (GSVM-RU) algorithm [12]. A variant of this GSVM technique has been successfully integrated into Secure Computing's TrustedSource reputation system for providing real-time

TABLE I
CONFUSION MATRIX

	predicted positives	predicted negatives
real positives	TP	FN
real negatives	FP	TN

collaborative sharing of global intelligence about the latest email threats [3]. However, it remains unclear how GSVM-RU performs compared to other state-of-the-art algorithms. Therefore, we present an exhaustive empirical study on benchmark datasets.

In this work, we also theoretically extend GSVM-RU based on the information loss minimization principle and design a new "combine" aggregation method. Furthermore, we revise it as a highly effective and efficient SVM modeling technique by explicitly executing granulation and aggregation by turns, and hence avoiding extracting too many negative granules. As a prior-knowledge-guided repetitive undersampling strategy to "rebalance" the dataset at hand, GSVM-RU can improve classification performance by 1) extracting informative samples that are essential for classification and 2) eliminating a large amount of redundant or even noisy samples. Besides GSVM-RU, we also propose three other SVM modeling methods that overweight the minority class, oversample the minority class, or undersample the majority class. These SVM modeling methods are compared favorably to previous works in 25 groups of experiments.

The rest of the paper is organized as follows. Background knowledge is briefly reviewed in Section II. Section III presents GSVM-RU and three other SVM modeling algorithms with different "rebalance" techniques. Section IV compares these four algorithms to state-of-the-art approaches on seven highly imbalanced datasets under different metrics. Finally, Section V concludes the paper.

II. BACKGROUND

A. Metrics for Imbalanced Classification

Many metrics have been used for effectiveness evaluation on imbalanced classification. All of them are based on the confusion matrix as shown at Table I. With highly skewed data distribution, the overall accuracy metric at (1) is not sufficient any more. For example, a naive classifier that predicts all samples as negative has high accuracy. However, it is totally useless to detect rare positive samples. To deal with class imbalance, two kinds of metrics have been proposed.

$$accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{1}$$

To get optimal balanced classification ability, sensitivity at (2) and specificity at (3) are usually adopted to monitor classification performance on two classes separately. Notice that sensitivity is also called true positive rate or positive class accuracy, while specificity called true negative rate or negative class accuracy. Based on these two metrics, G-Mean was proposed at (4), which is the geometric mean of sensitivity and specificity [13]. Furthermore, Area Under ROC Curve (AUC-ROC) can also indicate balanced classification ability between sensitivity and specificity as a function of varying a classification threshold [14].

$$sensitivity = TP/(TP + FN) \tag{2}$$

$$specificity = TN/(TN + FP) \tag{3}$$

$$G - Mean = \sqrt{sensitivity * specificity} \tag{4}$$

On the other hand, sometimes we are interested in highly effective detection ability for only one class. For example, for credit card fraud detection problem, the target is detecting fraudulent transactions. For diagnosing a rare disease, what we are especially interested in is to find patients with this disease. For such problems, another pair of metrics, precision at (5) and recall at (6), is often adopted. Notice that recall is the same as sensitivity. F-Measure at (7) is used to integrate precision and recall into a single metric for convenience of modeling [15]. Similar to AUC-ROC, Area Under Precision/Recall Curve (AUC-PR) can be used to indicate the detection ability of a classifier between precision and recall as a function of varying a decision threshold [16].

$$precision = TP/(TP + FP) \tag{5}$$

$$recall = TP/(TP + FN) \tag{6}$$

$$F - Measure = \frac{2 * precision * recall}{precision + recall} \tag{7}$$

In this work, the *perf* code, which is available at http://kodiak.cs.cornell.edu/kddcup/software.html, is utilized to calculate all of the four metrics.

B. Previous Methods for Imbalanced Classification

Many methods have been proposed for imbalanced classification, and some good results have been reported [2]. These methods can be categorized into three different categories: cost-sensitive learning, oversampling the minority class or undersampling the majority class. Interested readers may refer to [17] for a good survey. However, different measures have been used by different authors, which makes comparisons difficult.

Recently, several new models have been reported in the literature with good classification performance on imbalanced data. Hong et al. proposed a classifier construction approach based on orthogonal forward selection (OFS), which precisely aims at high effectiveness and efficiency [18]. Huang et al. proposed Biased Minimax Probability Machine (BMPM), which offers an elegant and systematic way to incorporate a certain bias for the minority class by directly controlling the lower bound of the real accuracy [19], [20].

Previous research that aims to improve the effectiveness of SVM on imbalanced classification includes the following. Vilariño et al. used Synthetic Minority Oversampling TEchnique (SMOTE) [21] oversampling and also random undersampling for SVM modeling on an imbalanced intestinal contractions detection task [22]. Raskutti et al. demonstrated that a one-class SVM that learned only from the minority

class sometimes can perform better than an SVM modeled from two classes [8]. Akbani et al. proposed the SMOTE with Different Costs algorithm (SDC) [6]. SDC conducts SMOTE oversampling on the minority class with different error costs. As a result, the decision boundary can be far away from the minority class. Wu et al. proposed the Kernel Boundary Alignment algorithm (KBA) that adjusts the boundary toward the majority class by modifying the kernel matrix [7].

Vilariño et al. worked on only one dataset [22]. One-class SVM actually performs worse in many cases compared to a standard two-class SVM [8]. SDC or KBA improves classification effectiveness on a two-class SVM. However, they are not efficient and hence difficult to be scalable to very large datasets. Wu et al. reported KBA usually takes a longer time for classification than SVM [7]. SDC is also slower than standard SVM modeling because oversampling increases the number of SVs. Unfortunately, SVM itself is already very slow on large datasets [9], [10], [11].

Our work contrasts with the previous work as follows:

- Most of prior works evaluate classification performance only on one or two metrics mentioned above. We present a broader experimental study on all four metrics.
- Most of previous works use decision trees as the basic classifier [1]. While, there are some recent papers on SVM for imbalanced classification [22], [8], [6], [7], the application of SVM is still not completely explored, particularly the realm of undersampling of SVs. Because SVM decides the class of a sample only based on SVs, which are training samples close to the decision boundary, the modeling effectiveness and efficiency may be improved for imbalanced classification by exploring the SVs-based undersampling.

III. GSVM-RU Algorithm

Granular computing represents information in the form of some aggregates (called information granules) such as subsets, subspaces, classes, or clusters of a universe. It then solves the targeted problem in each information granule [23]. There are two principles in granular computing. The first principle is divide-and-conquer to split a huge problem into a sequence of granules (granule split); The second principle is data cleaning to define the suitable size for one granule to comprehend the problem at hand without getting buried in unnecessary details (granule shrink). As opposed to traditional data-oriented numeric computing, granular computing is knowledge-oriented [24]. By embedding prior knowledge or prior assumptions into the granulation process for data modeling, better classification can be obtained. A granular computing-based learning framework called Granular Support Vector Machines (GSVM) was proposed in our previous work [25]. GSVM combines the principles from statistical learning theory and granular computing theory in a systematic and formal way. GSVM extracts a sequence of information granules with granule split and/or granule shrink, and then builds SVMs on some of these granules if necessary. The main potential advantages of GSVM are:

- GSVM is more sensitive to the inherent data distribution by establishing a trade-off between local significance of a subset of data and global correlation among different subsets of data, or between information loss and data cleaning. Hence, GSVM may improve classification effectiveness.
- GSVM may speed up the classification process by eliminating redundant data locally. As a result, it is more efficient and scalable on huge datasets.

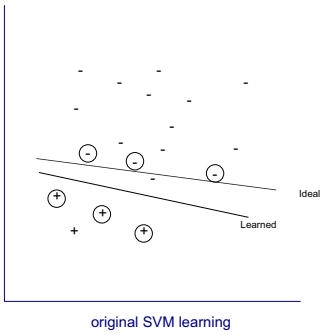


Fig. 1. Original SVM modeling. The circled points denote SVs.

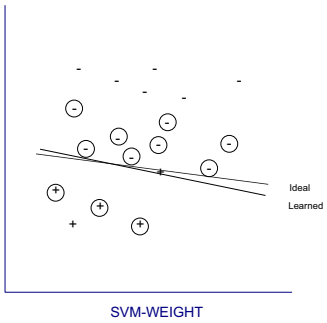


Fig. 2. SVM-WEIGHT modeling. The circled points denote SVs.

Based on GSVM, we propose *Granular Support Vector Machines - Repetitive Undersampling* (GSVM-RU) that is specifically designed for highly imbalanced classification.

A. GSVM-RU

SVM assumes that only SVs are informative to classification and other samples can be safely removed. However, for highly imbalanced classification, the majority class pushes the ideal decision boundary toward the minority class [6], [7]. As demonstrated in Fig. 1, negative SVs (the circled minus signs) that are close to the learned boundary may not be the most informative or even noisy. Some informative samples may hide behind them. To find these informative samples, we can conduct cost-sensitive learning or oversampling. However, these two "rebalance" strategies increase the number of SVs (Fig. 2 and Fig. 3), and hence slow down the classification process.

To improve efficiency, it is natural to decrease the size of the training dataset. In this sense, undersampling is by nature more suitable to model an SVM for imbalanced classification than other approaches. However, elimination of some samples from the training dataset may have two effects:

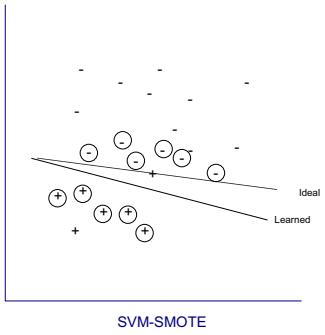


Fig. 3. SVM-SMOTE modeling. The circled points denote SVs.

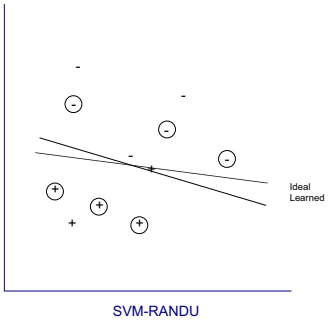


Fig. 4. SVM-RANDU modeling. The circled points denote SVs.

- Information loss: due to the elimination of informative or useful samples, classification effectiveness is deteriorated;
- Data cleaning: because of the elimination of irrelevant or redundant or even noisy samples, classification effectiveness is improved.

For a highly imbalanced dataset, there may be many redundant or noisy negative samples. Random undersampling is a common undersampling approach for rebalancing the dataset to achieve better data distribution. However, random undersampling suffers from information loss. As Fig. 4 shows, although random undersampling pushes the learned boundary close to the ideal boundary, the cues about the orientation of the ideal boundary may be lost [6].

GSVM-RU is targeted to directly utilize SVM itself for undersampling. The idea is based on the well-known fact about SVM - only SVs are necessary and other samples can be safely removed without affecting classification. This fact motivates us to explore the possibility to utilize SVM for data cleaning/undersampling.

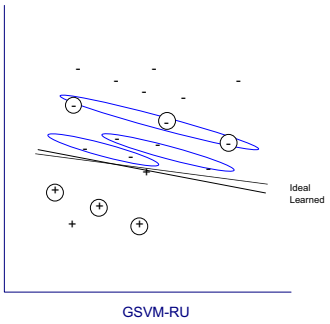


Fig. 5. GSVM-RU modeling. The circled points denote SVs.

However, due to highly skewed data distribution, the SVM modeled on the original training dataset is prone to classify every sample to be negative. As a result, a single SVM cannot guarantee to extract all informative samples as SVs. Fortunately, it seems reasonable to assume one single SVM can extract a part of, although not all, informative samples. Under this assumption, multiple information granules with different informative samples can be formed by following granulation operations. Firstly, we assume that all positive samples are informative to form a positive information granule. Secondly, negative samples extracted by an SVM as SVs are also possibly informative so that they form a negative information granule. Here we call these negative samples Negative Local Support Vectors (NLSVs). Then these NLSVs are removed from the original training dataset to generate a smaller training dataset, on which a new SVM is modeled to extract another group of NLSVs. This process is repeated several times to form multiple negative information granules. After that, all other negative samples still remaining in the training dataset are simply discarded.

An aggregation operation is then executed to selectively aggregate the samples in these negative information granules with all positive samples to complete the undersampling process. Finally, an SVM is modeled on the aggregated dataset for classification. As demonstrated in Fig. 5, because only a part of NLSVs (and the negative samples very far from the decision area) are removed from the original dataset, GSVM-RU undersampling can still give good cues about the orientation of the ideal boundary, and hence can overcome the shortcoming of random undersampling as mentioned above. Fig. 6 sketches the idea of GSVM-RU.

For SVM modeling, GSVM-RU adds another hyper-parameter Gr, the number of negative granules. To implement GSVM-RU as an utilizable algorithm, there are two related problems:

- How many negative information granules should be formed?
- How to aggregate the samples in these information granules?

It seems safe to extract more granules to reduce information loss. However, information contributed by two different granules may be redundant or even noisy to each other. And hence, less granules may decrease these redundancy or noise from the final aggregation dataset. In general, if Gr granules are extracted, we have 2^{Gr} different combinations to build the final aggregation dataset. It is extremely expensive to try all of these combinations.

For simplicity and efficiency, we revise the preliminary GSVM-RU algorithm [12] and propose to run granulation and aggregation in turns in this work. Firstly, the aggregation dataset is initialized to consist of only positive samples. And the best classification performance is initialized to be the performance of the naive classifier that classifies every sample as negative. When a new negative granule is extracted, the corresponding NLSVs are immediately aggregated into the

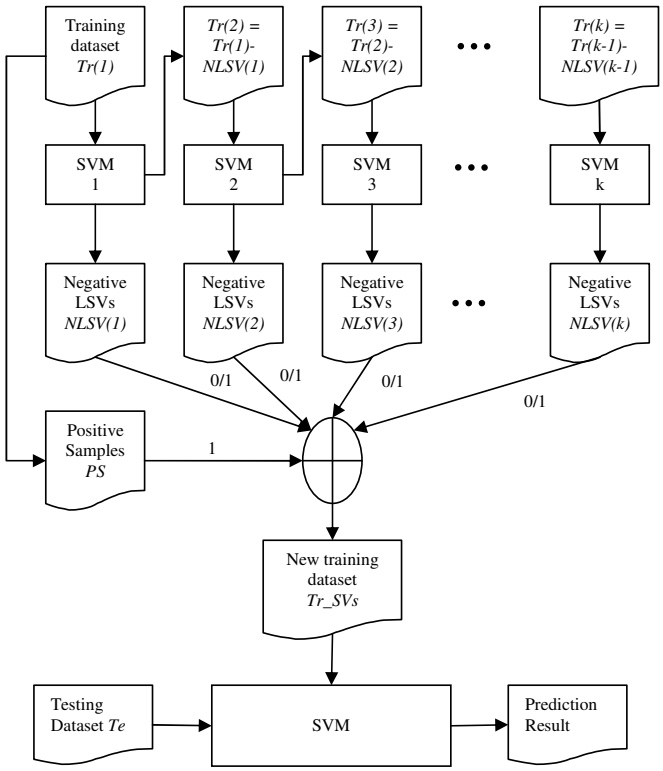


Fig. 6. Basic idea of GSVM-RU.

aggregation dataset. An SVM is then modeled on this new aggregation dataset. If classification performance is improved, we continue to the next phase to extract another granule. Otherwise the repetitive undersampling process is stopped and the classifier in the previous phase will be saved for future classification.

In [12], we proposed the "discard" operation for aggregation. When a new negative granule is extracted, only negative samples in the latest granule are aggregated into the new aggregation dataset and all samples in old negative granules are discarded. This operation is based on the "boundary push" assumption. If old NLSVs are discarded, the decision boundary is expected to be closer to the ideal one. The repetitive undersampling process is stopped when the new extracted granule alone cannot further improve classification performance.

However, the "discard" operation is not always suitable because it removes all previous negative granules, which are likely informative. In this work, we design a new "combine" aggregation operation. When a new granule is extracted, it is combined with all old granules to form a new aggregation dataset. The assumption is that not all informative samples can be extracted as NLSVs in one granule. As a result, it is expected to reduce information loss by extracting NLSVs multiple times. The repetitive undersampling process is stopped when the new extracted granule cannot further improve classification performance if joint with the previous aggregation dataset.

Which aggregation operation is better is data-dependent and also metric-dependent. For efficiency, we run both of them only when the second negative granule is extracted. The winner will be used for the following aggregation. All SVMs modeled in the repetitive process use the same kernel and the same parameters, which are tuned with grid search [26]. With such a repetitive undersampling process, a clear knowledge-oriented data cleaning strategy is implemented.

B. Three Other SVM Modeling Algorithms

In this research, we investigate three other "rebalance" techniques on SVM modeling for an exhaustive comparison study.

1) SVM-WEIGHT: SVM-WEIGHT implements cost-sensitive learning for SVM modeling. The basic idea is to assign a larger penalty value to FNs than FPs [27], [28], [6]. Although the idea is straightforward and has been implemented in LIBSVM [26], there is no systematic experimental report yet to evaluate the performance of this idea on highly imbalanced classification. Without loss of generality, the cost for a FP is always 1. The cost for a FN is usually suggested to be the ratio of negative samples over positive samples. However, our experiments show that it is not always optimal. And hence we add one parameter Rw into this algorithm. If there are Np positive samples and Nn negative samples, the FN cost should be $Nn/(Rw * Np)$. The optimal value of Rw is decided by grid search.

2) SVM-SMOTE: SVM-SMOTE adopts the SMOTE algorithm [21] to generate more pseudo positive samples and then builds an SVM on the oversampling dataset [6]. SVM-SMOTE also introduces one parameter Ro. If there are Np positive samples, we should add $Ro * Np$ pseudo positive samples into the original training dataset. The optimal value of Ro is decided by grid search.

3) SVM-RANDU: SVM-RANDU randomly selects a few of negative samples and then builds an SVM on the undersampling dataset [6]. Random undersampling were studied in [13], [1]. SVM-RANDU has an unknown parameter Ru. If there are Np positive samples, we should randomly select $Ru * Np$ negative samples from the original training dataset. The optimal value of Ru is decided by grid search.

C. Time Complexity

On highly imbalanced data, an SVM typically needs $O((Np+Nn)^3)$ time for training in the worst case [5]. SVM-RANDU takes $O((Np + Np * Ru)^3)$, which is faster than SVM because typically $Np * Ru << Nn$. SVM-SMOTE takes $O((Np * (Ro + 1) + Nn)^3)$, which is slower than SVM because it increases the size of the training dataset. SVM-WEIGHT seems to take the same $O((Np+Nn)^3)$ time as SVM. However, overweighting typically makes it harder for SVM learning to converge and hence it usually takes a longer time than SVM without overweighting. GSVM-RU takes $O(Gr*(Np+Nn)^3)$ because we need to model an SVM for each granule extraction. However, the later modeling steps are faster since the previous negative SVs (which are "hard" samples for classification) have been removed.

At the prediction phase, if an SVM has Ns SVs and there are Nu unknown samples for prediction, it takes $O(Ns*Nu)$ time for prediction. Our experiments demonstrate that GSVM-RU and SVM-RANDU extracts significantly less SVs and hence are more efficient.

IV. Empirical Studies

The experiments are conducted on a machine with a centrino-1.6MHz CPU and 1024M memory. The software is based on the OSU SVM Classifier Matlab Toolbox, which is available at http://sourceforge.net/projects/svm/ and implements a Matlab interface to LIBSVM [26].

A. Data Sets

Seven datasets, collected from related works, are used in our empirical studies. As shown in Table II, all of them are highly imbalanced as less than 10% samples are positive. There are also significant variations of the data size (from several hundreds to over ten thousands) and the number of features (from 6 to 49).

For each dataset, the performance is evaluated with four metrics: G-Mean, AUC-ROC, F-Measure, and AUC-PR.

The classification performance is estimated with different training/testing heuristics. For 5 of the 7 datasets, 10-fold Cross Validation is used. For Abalone (19 vs. other) and Yeast (ME2 vs. other) datasets, it is estimated by averaging on 7 times random partition with the training/testing ratio 6:1 or 7:3. Basically, if a training/testing heuristic was used for a dataset in previous works, we also use it for comparison.

For each fold or each training/testing process, firstly the data is normalized so that each input feature has 0 mean and 1 standard deviation on the training dataset; then classification algorithms are executed on the normalized training dataset and the model parameters are optimized by grid search.

The modeling process is carried out separately for each of the four metrics. SVM-SMOTE and SVM-RANDU are executed 10 times and the *average performance*$\pm$*standard deviation* is reported. SVM-WEIGHT and GSVM-RU are executed only once because they are stable in the sense that the performance is never modified among multiple runs if parameters are fixed.

In the following, only high level comparisons between GSVM-RU and other approaches are reported. Readers can access detailed comparison results on each dataset at http://tinman.cs.gsu.edu/∼cscyntx/gsvm-ru/imbalance-result.pdf.

B. How GSVM-RU improves classification

With limited space, the mammography dataset is used as one example to show how GSVM-RU works to improve classification. We obtain similar performance gains on other datasets.

With G-Mean for evaluation, the best validation performance is observed when the "discard" aggregation operation is adopted and the 4th granule is used as the final aggregation dataset. (i.e., The first three granules are discarded). The result

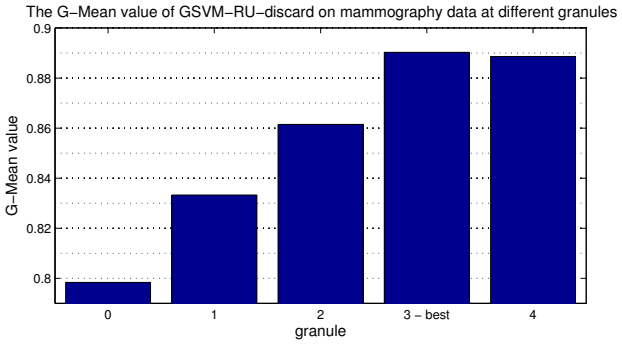


Fig. 7. G-Mean values of GSVM-RU modeling with "discard" operation on mammography data.

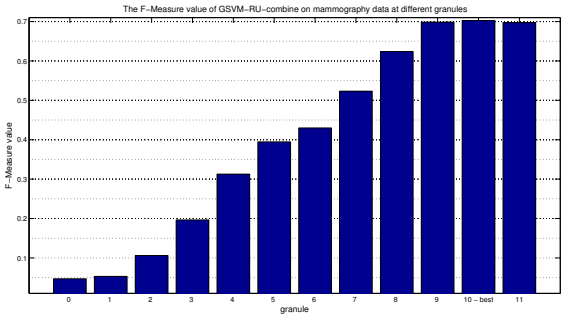


Fig. 8. F-Measure values of GSVM-RU modeling with "combine" operation on mammography data.

indicates that the first assumption (the decision boundary is pushed toward the minority class) is reasonable here. When the NLSVs in the old granules are discarded, the decision boundary gradually goes back to the "ideal" one and thus classification performance is improved (Fig. 7). After the 5th granule is extracted, too many informative samples are discarded so that classification performance is deteriorated. And hence the repetitive undersampling process is stopped.

With F-Measure for evaluation, the best validation performance is observed when the "combine" aggregation operation is adopted and the first 11 granules are combined to form the final aggregation dataset. The result indicates that the second assumption (a part but not all of informative samples can be extracted in one granule) is reasonable in this case. When more and more informative samples are combined into the aggregated dataset, information loss is less and less so that more accurate classification can be obtrained (Fig. 8). However, when the 12th granule is extracted and combined into the aggregation dataset, the validation performance can not be further improved. The reason is that the new extracted samples are too far from the "ideal" boundary so that they are prone to be redundant or irrelevant other than informative. And hence the repetitive undersampling process is stopped.

C. GSVM-RU vs. Previous Best Approaches

25 groups of experiments are conducted with 25 different dataset/metric combinations (Table III). Of them 18 groups are available for effectiveness comparison with previous studies. For 12 groups, GSVM-RU outperforms the previous best

TABLE II
CHARACTERISTICS OF DATASETS

Dataset	# of Samples	# of Positive samples (%)	# of Features	Validation Method
Oil	937	41 (4.38%)	49	10-fold CV
Mammography	11183	260 (2.32%)	6	10-fold CV
Satimage	6435	626 (9.73%)	36	10-fold CV
Abalone (19 vs. other)	4177	32 (0.77%)	8	6:1 or 7:3 partition
Abalone (9 vs. 18)	731	42 (5.75%)	8	10-fold CV
Yeast (ME2 vs. other)	1484	51 (3.44%)	8	6:1 partition
Yeast (CYT vs. POX)	483	20 (4.14%)	8	10-fold CV

TABLE IV
AVERAGE PERFORMANCE OF PREVIOUS BEST APPROACHES AND GSVM-RU ON 18 EXPERIMENTS

	previous best	GSVM-RU
G-Mean/8	79.7	85.2
AUC-ROC/5	90.2	92.4
F-Measure/5	59.9	66.5

approach. For 6 other groups, the performance of GSVM-RU is very close to the previous best result.

Table IV reports the average performance on G-Mean, AUC-ROC and F-Measure metrics on the 18 groups of experiments. GSVM-RU demonstrates better average performance than previous best approaches on all three metrics.

Figures 9(a)-10(b) visualize comparison results on the four metrics, respectively. In each figure, performance of the previous best approach, SVM-WEIGHT, SVM-SMOTE, SVM-RANDU and GSVM-RU is reported for each available dataset. The value on the horizontal axis is formatted as *data-name(previous-best-approach-name)*. If there is no previous result, only *data-name* is reported. It can be clearly seen that GSVM-RU and the other 3 SVM modeling algorithms are able to surpass or match the previously known best algorithms on each of the 18 dataset/metric combinations. That is, we effectively compare these SVM modeling techniques against the best known approaches under the same experimental conditions. Notice that in Fig. 9(a), the G-mean value of SVM-SMOTE is 0 for the abalone dataset (19 vs. other) with both 7 times 6:1 splitting or 7 times 7:3 splitting. Also notice that there are no "previous best" results in Fig. 10(b) because no previous research has conducted an AUC-PR analysis on these datasets.

D. GSVM-RU vs. Other 3 SVM Modeling Algorithms

Table V reports the average performance of four SVM modeling algorithms over the 25 groups of experiments. SVM-WEIGHT demonstrates almost the same effectiveness as GSVM-RU with all four metrics. However, GSVM-RU extracts only 181 SVs, which means that GSVM-RU is more than 4 times faster than SVM-WEIGHT (with 794 SVs) for classification.

SVM-SMOTE demonstrates similar effectiveness on AUC-ROC, F-Measure and AUC-PR to GSVM-RU. However, it is worse on G-Mean. The reason is that it achieves 0 G-Mean value on the extremely imbalanced abalone (19 vs. other) dataset. SVM-SMOTE is also much slower with 655 SVs for classification. Moreover, SVM-SMOTE is unstable because of the randomness of the oversampling process.

SVM-RANDU is slightly faster than GSVM-RU for prediction by extracting only 143 SVs. However, SVM-RANDU is slightly less effective than GSVM-RU with all four metrics. Moreover, SVM-RANDU is unstable because of the randomness of the undersampling process.

V. CONCLUSIONS

We implement and rigorously evaluate four SVM modeling techniques, including one novel method of undersampling SVs, for our work. We compare these four algorithms with state-of-the-art approaches on seven highly imbalanced datasets under four metrics (G-Mean, AUC-ROC, F-Measure, and AUC-PR). The comparative approaches consist of the best known technique on the corresponding datasets. As far as we know, this is the first work to conduct an exhaustive comparative study with all four metrics and the variations in SVM modeling. And hence we expect that our work can be helpful for future research works to do comparison study on these benchmark highly imbalanced datasets.

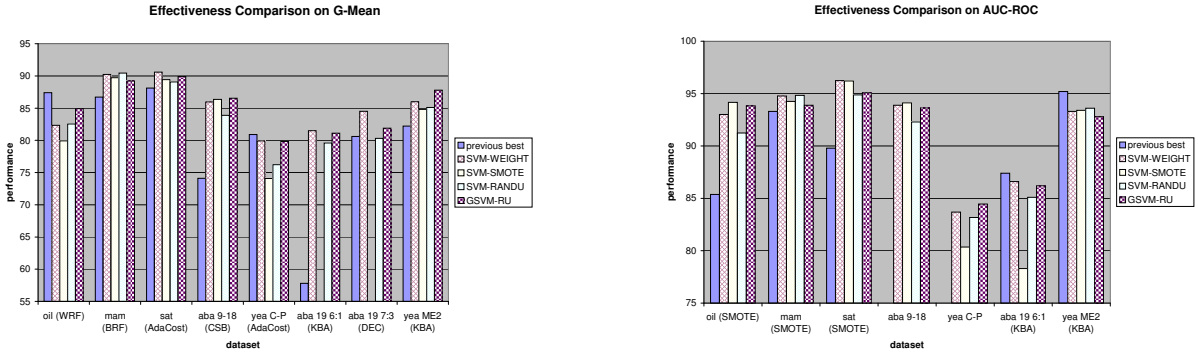
Specifically, the *Granular Support Vector Machines - Repetitive Undersampling* algorithm implements a guided repetitive undersampling strategy to "rebalance" the dataset at hand. GSVM-RU is effective due to 1) extraction of informative samples that are essential for classification and 2) elimination of a large amount of redundant or even noisy samples. As shown in Table III, GSVM-RU outperforms the previous best approach in 12 groups of experiments, and performs very close to the previous best approach in other 6 groups of experiments.

In most of cases, GSVM-RU achieves the optimal performance with the "discard" operation. This demonstrates that the "boundary push" assumption seems to be true for many highly imbalanced datasets. Considering its efficiency, the "discard" operation is also suggested as the first aggregation operation to try for GSVM-RU modeling. However, the optimal performance is observed with the "combine" operation on the mammography dataset and the satimage dataset for F-Measure and AUC-PR metrics. This suggests that the "information loss" assumption may be more suitable for some highly imbalanced datasets, especially with F-Measure and AUC-PR metrics.

We also systematically investigate the effect of overweighting the minority class on SVM modeling. The idea, named SVM-WEIGHT, seems to be naive at first glance and hence is ignored by previous research works. However, our experiments show that it is actually highly effective. Although SVM-WEIGHT is not efficient compared to GSVM-RU since it

TABLE III
EFFECTIVENESS OF CLASSIFICATION

dataset	metric	validation	previous best	GSVM-RU	best in this work
Oil	G-Mean	10-fold CV	**87.4** (WRF [29])	84.9 (D)	84.9 (GSVM-RU)
Mam	G-Mean	10-fold CV	86.7 (BRF [29])	**89.0** (D)	**90.5** (SVM-RANDU)
Sat	G-Mean	10-fold CV	88.1 (AdaCost [30])	**89.9** (D)	**90.6** (SVM-WEIGHT)
A(9-18)	G-Mean	10-fold CV	74.1 (CSB2 [30])	**86.5** (D)	**86.5** (GSVM-RU)
Y(C-P)	G-Mean	10-fold CV	**80.9** (AdaCost [30])	79.8 (D)	79.9 (SVM-WEIGHT)
A(19)	G-Mean	7 times 6:1	57.8 (KBA [7])	**81.1** (D)	**81.5** (SVM-WEIGHT)
A(19)	G-Mean	7 times 7:3	80.6 (DEC [6])	**81.9** (D)	**84.5** (SVM-WEIGHT)
Y(ME2)	G-Mean	7 times 6:1	82.2 (KBA [7])	**87.8** (D)	**87.8** (GSVM-RU)
Oil	AUC-ROC	10-fold CV	85.4 (SMOTE [21])	**93.8** (D)	**94.2** (SVM-SMOTE)
Mam	AUC-ROC	10-fold CV	93.3 (SMOTE [21])	**93.9** (D)	**94.8** (SVM-RANDU)
Sat	AUC-ROC	10-fold CV	89.8 (SMOTE [21])	**95.1** (D)	**96.2** (SVM-WEIGHT)
A(9-18)	AUC-ROC	10-fold CV	N/A	93.6 (D)	94.1 (SVM-SMOTE)
Y(C-P)	AUC-ROC	10-fold CV	N/A	84.5 (D)	84.5 (GSVM-RU)
A(19)	AUC-ROC	7 times 6:1	**87.4** (KBA [7])	86.2 (D)	86.6 (SVM-WEIGHT)
Y(ME2)	AUC-ROC	7 times 6:1	**95.2** (KBA [7])	92.8 (D)	93.6 (SVM-RANDU)
Oil	F-Measure	10-fold CV	55.0 (DataBoost-IM [30])	**64.1** (D)	**66.7** (SVM-WEIGHT)
Mam	F-Measure	10-fold CV	**71.3** (WRF [29])	70.2 (C)	70.2 (GSVM-RU)
Sat	F-Measure	10-fold CV	**70.2** (SMOTE-Boost [31])	69.1 (C)	69.7 (SVM-SMOTE)
A(9-18)	F-Measure	10-fold CV	45.0 (DataBoost-IM [30])	**60.4** (D)	**64.7** (SVM-SMOTE)
Y(C-P)	F-Measure	10-fold CV	58.0 (DataBoost-IM [30])	**68.8** (D)	**68.8** (ALL)
Oil	AUC-PR	10-fold CV	N/A	58.8 (D)	61.1 (SVM-WEIGHT)
Mam	AUC-PR	10-fold CV	N/A	64.3 (C)	68.4 (SVM-SMOTE)
Sat	AUC-PR	10-fold CV	N/A	74.4 (C)	75.4 (SVM-SMOTE)
A(9-18)	AUC-PR	10-fold CV	N/A	65.5 (D)	66.6 (SVM-WEIGHT)
Y(C-P)	AUC-PR	10-fold CV	N/A	62.9 (D)	62.9 (GSVM-RU)

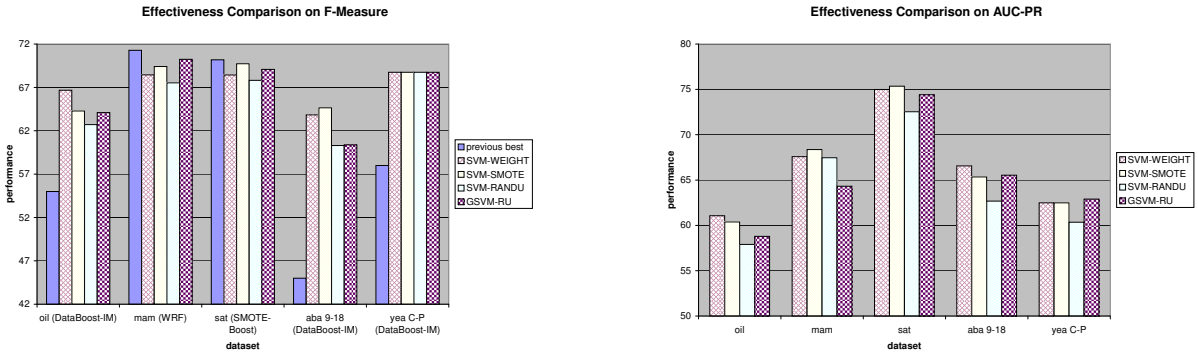


(a) G-Mean (b) AUC-ROC

Fig. 9. G-Mean and AUC-ROC analysis.

TABLE V
AVERAGE PERFORMANCE OF FOUR SVM MODELING ALGORITHMS ON 25 EXPERIMENTS

	SVM-WEIGHT	SVM-SMOTE	SVM-RANDU	GSVM-RU
G-Mean/8	85.1	63.0	83.4	85.2
AUC-ROC/7	91.6	90.1	90.7	91.4
F-Measure/5	67.2	67.4	65.4	66.5
AUC-PR/5	66.5	66.4	64.2	65.2
Efficiency/25 (#SVs)	794	655	143	181
Stability	YES	NO	NO	YES



(a) F-Measure

(b) AUC-PR

Fig. 10. F-Measure and AUC-PR analysis.

extracts more SVs, it can be the first SVM modeling method of choice if the available dataset is not very large.

REFERENCES

[1] N. Japkowicz and S. Stephen, "The class imbalance problem: A systematic study," *Intelligent Data Analysis*, vol. 6, no. 5, pp. 429–449, 2002.

[2] N. V. Chawla, N. Japkowicz, and A. Kotcz, "Editorial: special issue on learning from imbalanced data sets," *SIGKDD Explorations*, vol. 6, no. 1, pp. 1–6, 2004.

[3] Y. C. Tang, S. Krasser, P. Judge, and Y.-Q. Zhang, "Fast and effective spam IP detection with granular SVM for spam filtering on highly imbalanced spectral mail server behavior data," in *Proc. of The 2nd International Conference on Collaborative Computing*, 2006.

[4] V. N. Vapnik, *Statistical Learning Theory*. New York: John Wiley and Sons, 1998.

[5] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[6] R. Akbani, S. Kwek, and N. Japkowicz, "Applying support vector machines to imbalanced datasets," in *Proc. of the 15th European Conference on Machine Learning (ECML 2004)*, pp. 39–50, 2004.

[7] G. Wu and E. Y. Chang, "KBA: Kernel boundary alignment considering imbalanced data distribution," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 786–795, 2005.

[8] B. Raskutti and A. Kowalczyk, "Extreme re-balancing for SVMs: a case study," *SIGKDD Explorations*, vol. 6, no. 1, pp. 60–69, 2004.

[9] J. Dong, C. Y. Suen, and A. Krzyzak, "Algorithms of fast SVM evaluation based on subspace projection," in *Proc. of IJCNN '05, Vol. 2*, pp. 865–870, 2005.

[10] H. Isozaki and H. Kazawa, "Efficient Support Vector Classifiers for Named Entity Recognition," in *Proc. of the 19th International Conference on Computational Linguistics (COLING'02)*, pp. 390–396, 2002.

[11] B. L. Milenova, J. S. Yarmus, and M. M. Campos, "SVM in oracle database 10g: removing the barriers to widespread adoption of support vector machines," in *Proc. of the 31st international conference on Very large data bases*, pp. 1152–1163, 2005.

[12] Y. C. Tang and Y.-Q. Zhang, "Granular SVM with repetitive undersampling for highly imbalanced protein homology prediction," in *Proc. of GrC-IEEE 2006*, pp. 457–461, 2006.

[13] M. Kubat and S. Matwin, "Addressing the curse of imbalanced training sets: one-sided selection," in *Proc. of the 14th International Conference on Machine Learning (ICML1997)*, pp. 179–186, 1997.

[14] A. P. Bradley, "The use of the area under the ROC curve in the evaluation of machine learning algorithms.," *Pattern Recognition*, vol. 30, no. 7, pp. 1145–1159, 1997.

[15] C. J. Van Rijsbergen, *Information Retrieval, 2nd edition*. London: Butterworths, 1979.

[16] J. Davis and M. Goadrich, "The relationship between Precision-Recall and ROC curves," in *Proc. of the 23rd International Conference on Machine Learning (ICML 2006)*, pp. 233–240, 2006.

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

[18] X. Hong, S. Chen, and C. Harris, "A kernel-based two-class classifier for imbalanced data sets," *IEEE Transactions on Neural Networks*, vol. 18, no. 1, pp. 28–41, 2007.

[19] K. Huang, H. Yang, I. King, and M. R. Lyu, "Imbalanced learning with a biased minimax probability machine," *IEEE Transactions on System, Man, and Cybernetics Part B*, vol. 36, no. 4, pp. 913–923, 2006.

[20] K. Huang, H. Yang, I. King, and M. R. Lyu, "Maximizing sensitivity in medical diagnosis using biased minimax probability machine.," *IEEE Transactions on Biomedical Engineering*, vol. 53, no. 5, pp. 821–831, 2006.

[21] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence and Research*, vol. 16, pp. 321–357, 2002.

[22] F. Vilariño, P. Spyridonos, J. Vitrià, and P. Radeva, "Experiments with svm and stratified sampling with an imbalanced problem: Detection of intestinal contractions," in *Proc. of the 3rd International Conference on Advances in Pattern Recognition (ICAPR 2005)*, pp. 783–791, 2005.

[23] T. Y. Lin, "Data mining and machine oriented modeling: A granular computing approach," *Applied Intelligence*, vol. 13, no. 2, pp. 113–124, 2000.

[24] A. Bargiela and W. Pedrycz, *Granular Computing: An Introduction*. Kluwer: Kluwer Academic Pub, 2002.

[25] Y. C. Tang, B. Jin, and Y.-Q. Zhang, "Granular support vector machines with association rules mining for protein homology prediction," *Artificial Intelligence in Medicine*, vol. 35, no. 1-2, pp. 121–134, 2005.

[26] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001.

[27] E. Osuna, R. Freund, and F. Girosi, "Support vector machines: Training and applications," tech. rep., Massachusetts Institute of Technology, Cambridge, MA, USA, 1997.

[28] K. Veropoulos, N. Cristianini, and C. Campbell, "Controlling the sensitivity of support vector machines," in *Joint Conference on Artificial Intelligence (IJCAI)*, pp. 55–60, 1999.

[29] C. Chen, A. Liaw, and L. Breiman, "Using random forest to learn imbalanced data," tech. rep., Department of Statistics, UC Berkeley, Berkeley, CA, 2004.

[30] H. Guo and H. L. Viktor, "Learning from imbalanced data sets with boosting and data generation: the DataBoost-IM approach," *SIGKDD Explorations*, vol. 6, no. 1, pp. 30–39, 2004.

[31] N. V. Chawla, A. Lazarevic, L. O. Hall, and K. W. Bowyer, "SMOTEBoost: Improving prediction of the minority class in boosting," in *Proc. of the 7th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pp. 107–119, 2003.