Learning on the Border: Active Learning in Imbalanced Data Classification

Şeyda Ertekin[1], Jian Huang[2], Léon Bottou[3], C. Lee Giles[2,1]

[1]Department of Computer Science and Engineering
[2]College of Information Sciences and Technology
The Pennsylvania State University
University Park, PA 16802, USA
[3]NEC Laboratories America
4 Independence Way, Princeton, NJ 08540, USA
sertekin@cse.psu.edu, {jhuang, giles}@ist.psu.edu, leon@bottou.org

ABSTRACT

This paper is concerned with the class imbalance problem which has been known to hinder the learning performance of classification algorithms. The problem occurs when there are significantly less number of observations of the target concept. Various real-world classification tasks, such as medical diagnosis, text categorization and fraud detection suffer from this phenomenon. The standard machine learning algorithms yield better prediction performance with balanced datasets. In this paper, we demonstrate that active learning is capable of solving the class imbalance problem by providing the learner more balanced classes. We also propose an efficient way of selecting informative instances from a smaller pool of samples for active learning which does not necessitate a search through the entire dataset. The proposed method yields an efficient querying system and allows active learning to be applied to very large datasets. Our experimental results show that with an early stopping criteria, active learning achieves a fast solution with competitive prediction performance in imbalanced data classification.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: Miscellaneous; I.2.6 [**Artificial Intelligence**]: Learning—*concept learning, induction*

General Terms

Algorithms, experimentation

Keywords

Active learning, imbalanced data, support vector machines

1. INTRODUCTION

Classification is a supervised learning method which acquires a training dataset to form its model for classifying unseen examples. A training dataset is called imbalanced if at least one of the classes are represented by significantly less number of instances (i.e. observations, examples, cases) than the others. Real world applications often face this problem because naturally normal examples which constitute the majority class in classification problems are generally abundant; on the other hand the examples of interest are generally rare and form the minority class. Another reason for class imbalance problem is the limitations (e.g., cost, difficulty or privacy) on collecting instances of some classes. Examples of applications which may have class imbalance problem include, but are not limited to, predicting pre-term births [8], identifying fraudulent credit card transactions [4], text categorization [7], classification of protein databases [19] and detecting certain objects from satellite images [13]. Despite that they are difficult to identify, rare instances generally constitute the target concept in classification tasks. However, in imbalanced data classification, the class boundary learned by standard machine learning algorithms can be severely skewed toward the target class. As a result, the false-negative rate can be excessively high.

In classification tasks, it is generally more important to correctly classify the minority class instances. In real-world applications mispredicting a rare event can result in more serious consequences than mispredicting a common event. For example in the case of cancerous cell detection, misclassifying non-cancerous cells leads to additional clinical testing but misclassifying cancerous cells leads to very serious health risks. Similar problem might occur in detection of a threatening surveillance event from video streams, where misclassifying a normal event may only result in increased security but misclassifying a life threatening event may lead to disastrous consequences. However in classification problems with imbalanced data, the minority class examples are more likely to be misclassified than the majority class examples. Due to their design principles, most of the machine learning algorithms optimize the overall classification accuracy hence sacrifice the prediction performance on the minority classes. This paper proposes an efficient active learning framework which has high prediction performance to overcome this serious data mining problem.

In addition to the naturally occurring class imbalance problem, the imbalanced data situation may also occur in one-against-rest schema in multiclass classification. Assuming there are N different classes, one of the simplest multiclass classification schemes built on top of binary classifiers is to train N different binary classifiers. Each classifier is trained to distinguish the examples in a single class from the examples in all remaining classes. When it is desired to classify a new example, the N classifiers are run, and the classifier which has the highest classification confidence is chosen. Therefore, even though the training data is balanced, issues related to the class imbalance problem can frequently surface.

In this paper we propose an alternative to the existing methods: using active learning strategy to deal with the class imbalance problem. Active learning has been pronounced by some researchers [18, 1] as a sampling method but no systematic study has been done to show that it works well with imbalanced data. We demonstrate that by selecting informative instances for training, active learning can indeed be a useful technique to address the class imbalance problem. We constrain our discussion to a standard two-class classification problem with Support Vector Machines (SVMs). In the rest of the paper, we refer to the minority and majority classes as "positive" and "negative" respectively.

In this paper, we propose an efficient SVM based active learning selection strategy which queries small pool of data at each iterative step instead of querying the entire dataset. The proposed method brings the advantage of efficient querying in search of the most informative instances, thus enabling active learning strategy to be applied to large datasets without high computational costs. Rather than using a traditional batch SVM, we use an *online* SVM algorithm [3] which suits better to the nature of active learning due to its incremental learning steps. We present that active learning's querying strategy yields a balanced training set in the early stages of the learning without any requirement of preprocessing of the data. Major research direction in recent literature to overcome the class imbalance problem is to resample the original training dataset to create more balanced classes. This is done either by oversampling the minority class and/or undersampling the majority class until the classes are approximately equally represented. Our empirical results show that active learning can be a more efficient alternative to resampling methods in creating balanced training set for the learner. AL does not risk losing information as in undersampling and does not bring an extra burden of data as in oversampling. With early stopping, active learning can achieve faster and scalable solution without sacrificing prediction performance.

2. RELATED WORK

Recent research on class imbalance problem has focused on several major groups of techniques. One is to assign distinct costs to the classification errors [6, 17]. In this method, the misclassification penalty for the positive class is assigned a higher value than that of the negative class. This method requires tuning to come up with good penalty parameters for the misclassified examples. The second is to resample the original training dataset, either by over-sampling the minority class and/or under-sampling the majority class until the classes are approximately equally represented [5, 11, 14, 15]. Both resampling methods introduce additional computational costs of data preprocessing and oversampling can be overwhelming in the case of very large scale training data. Undersampling has been proposed as a good means of increasing the sensitivity of a classifier. However this method may discard potentially useful data that could be important for the learning process therefore significant decrease in the prediction performance may be observed. Discarding the redundant examples in undersampling has been discussed in [16] but since it is an adaptive method for ensemble learning and does not involve an external preprocessing step it can not be applied to other types of algorithms. Oversampling has been proposed to create synthetic

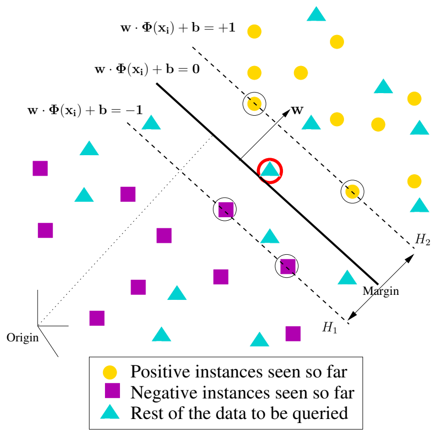

Figure 1: Active Learning with SVM (separable case). The most informative sample among the unseen training samples is the one (in bold circle) closest to the hyperplane (solid line). The circled samples on the dashed lines are support vectors.

positive instances from the existing positive samples to increase the representation of the class. Nevertheless, oversampling may suffer from overfitting and due to the increase in the number of samples, the training time of the learning process gets longer. If a complex oversampling method is used, it also suffers from high computational costs during preprocessing data. In addition to those, oversampling methods demand more memory space for the storage of newly created instances and the data structures based on the learning algorithm (i.e., extended kernel matrix in kernel classification algorithms). Deciding on the oversampling and undersampling rate is also another issue of those methods. Another technique suggested for class imbalance problem is to use a recognition-based, instead of discrimination-based inductive learning [10, 20]. These methods attempt to measure the amount of similarity between a query object and the target class, where classification is accomplished by imposing a threshold on the similarity measure. The major drawback of those methods is the need for tuning the similarity threshold of which the success of the method mostly relies on. On the other hand, discrimination-based learning algorithms have been proved to give better prediction performance in most domains.

In [2] the behavior of Support Vector Machines (SVM) with imbalanced data is investigated. They applied [5]'s SMOTE algorithm to oversample the data and trained SVM with different error costs. SMOTE is an oversampling approach in which the minority class is oversampled by creating synthetic examples rather than with replacement. The k nearest positive neighbors of all positive instances are identified and synthetic positive examples are created and placed randomly along the line segments joining the k minority class nearest neighbors. Preprocessing the data with SMOTE may lead to improved prediction performance at the classifiers, however it also brings more computational cost to the system for preprocessing and yet the increased number of training data makes the SVM training very costly since the training time at SVMs scales quadratically with the number of training instances. In order to cope with today's tremendously growing dataset sizes, we believe that there is a need for more computationally efficient and scalable algorithms. We show that such a solution can be achieved by using active learning strategy.

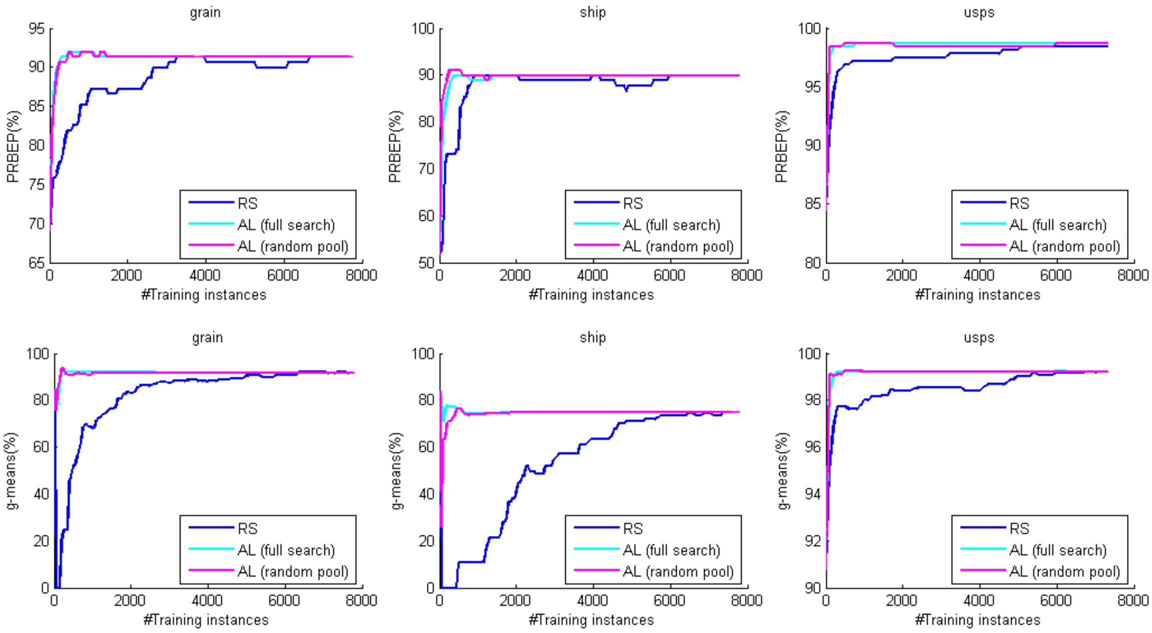

Figure 2: Comparison of PRBEP and g-means of RS, AL(full search) and AL(random pool). The training times of AL(full search) vs. AL(random pool) until saturation in seconds are: 272 vs. 50 (grain), 142 vs. 32 (ship) and 126 vs. 13 (USPS). AL(random pool) is 4 to 10 times faster than AL(full search) with similar prediction performance.

3. METHODOLOGY

Active learning is mostly regarded as a technique that addresses the unlabeled training instance problem. The learner has access to a vast pool of unlabeled examples, and it tries to make a clever choice to select the most informative example to obtain its label. However, in the cases where all the labels are available beforehand, active learning can still be leveraged to obtain the informative instances through the training sets [21, 3, 9]. In SVMs, *informativeness* of an instance is synonymous with its distance to the hyperplane. The farther an instance is to the hyperplane, the more the learner is confident about its true class label, hence it does not bring much (or any) information to the system. On the other hand, the instances close to the hyperplane are informative for learning. SVM based active learning can pick up the informative instances by checking their distances to the hyperplane. The closest instances to the hyperplane are considered to be the most informative instances.

The strategy of selecting instances within the margin addresses the imbalanced dataset classification very well. Suppose that the class distributions of an imbalanced dataset is given in Figure 3. The shaded region corresponds to the class distribution of the data within the margin. As it can be observed, the imbalance ratio of the classes within the margin is much smaller than the class imbalance ratio of the entire dataset. Any selection strategy which focuses on the instances in the margin most likely ends up with a more balanced class distribution than that of the entire dataset. Our empirical findings with various type of real-world data confirm that the imbalance ratios of the classes within the margin in real-world data are generally much lower than that of the entire data as shown in Figure 3.

A brief explanation of the SVMs is given in Section 3.1 followed by the working principles of the efficient active learning algorithm in Section 3.2. We explain the advantage of using online SVMs with the active sample selection in Section 3.3. In Section 3.4, we then describe an early stopping heuristics for active learning.

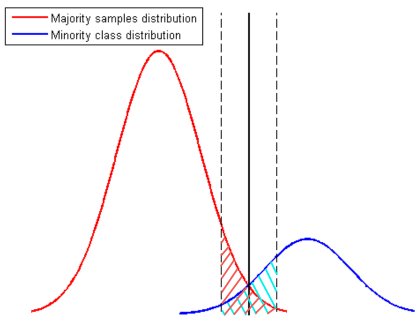

Figure 3: Data within the margin is less imbalanced than the entire data.

3.1 Support Vector Machines

Support Vector Machines [24] are well known for their strong theoretical foundations, generalization performance and ability to handle high dimensional data. In the binary classification setting, let $((x_1, y_1) \cdots (x_n, y_n))$ be the training dataset where x_i are the feature vectors representing the instances and $y_i \in (-1, +1)$ be the labels of the instances. Using the training set, SVM builds an optimum hyperplane – a linear discriminant in a higher dimensional feature space – that separates the two classes by the largest margin (see Figure 1). This hyperplane can be obtained by minimizing the following objective function:

$$\min_{\mathbf{w},\mathbf{b},\xi_\mathbf{i}} \frac{1}{2}\mathbf{w} \cdot \mathbf{w}^T + C\sum_{i=1}^{N}\xi_i \qquad (1)$$

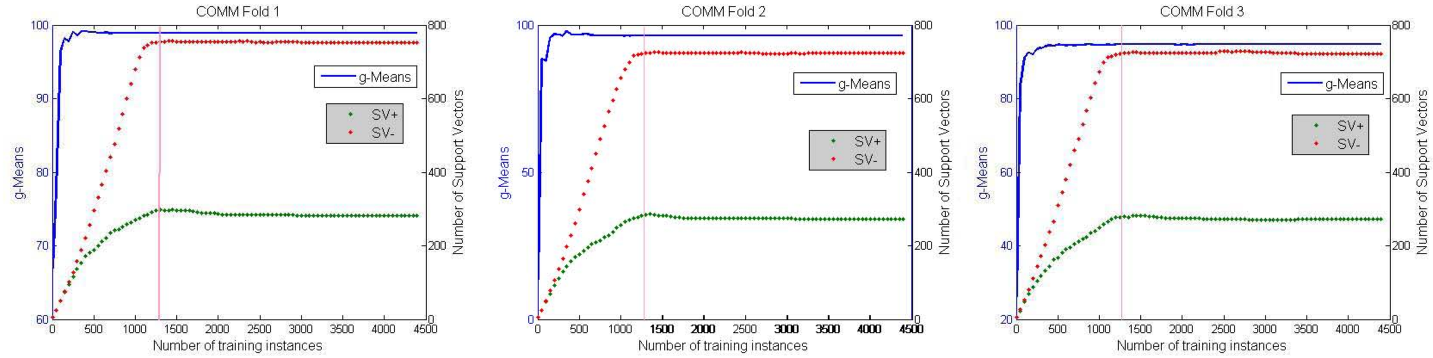

Figure 4: 3-fold cross-validation results for the training set of the category COMM in CiteSeer dataset. Vertical lines correspond to early stopping points.

$$\text{subject to} \begin{cases} \forall i \; y_i(\mathbf{w}^T\Phi(x_i) - b) \geq 1 - \xi_i \\ \forall i \; \xi_i \geq 0 \end{cases} \tag{2}$$

where $\mathbf{w}$ is the norm of the hyperplane, b is the offset, y_i are the labels, $\Phi(\cdot)$ is the mapping from input space to feature space, and ξ_i are the slack variables that permit the non-separable case by allowing misclassification of training instances. In practice the convex quadratic programming (QP) problem in Equation 1 is solved by optimizing the dual cost function. The dual representation of Equation 1 is given as

$$\max W(\alpha) \equiv \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j K(\mathbf{x_i}, \mathbf{x_j}) \tag{3}$$

$$\text{subject to} \begin{cases} \forall i \; 0 \leq \alpha_i \leq C \\ \sum_{i=1}^{N} \alpha_i y_i = 0 \end{cases} \tag{4}$$

where y_i are the labels, $\Phi(\cdot)$ is the mapping from the input space to the feature space, $K(\mathbf{x_i}, \mathbf{x_j}) = \langle \Phi(\mathbf{x_i}), \Phi(\mathbf{x_j}) \rangle$ is the kernel matrix and the α_i's are the *Lagrange multipliers* which are non-zero only for the training instances which fall in the margin. Those training instances are called *support vectors* and they define the position of the hyperplane. After solving the QP problem, the norm of the hyperplane $\mathbf{w}$ can be represented as

$$w = \sum_{i=1}^{n} \alpha_i \Phi(x_i) \tag{5}$$

3.2 Active Learning

Note that in equation 5, only the support vectors have an effect on the SVM solution. This means that if SVM is retrained with a new set of data which only consist of those support vectors, the learner will end up finding the same hyperplane. This fact leads us to the idea that not all the instances are equally important in the training sets. Then the question is how to select the most informative examples in the datasets. In this paper we will focus on a form of selection strategy called SVM based active learning. In SVMs, the most informative instance is believed to be the closest instance to the hyperplane since it divides the *version space* into two equal parts. The aim is to reduce the version space as fast as possible to reach the solution faster in order to avoid certain *costs* associated with the problem. For the possibility of a non-symmetric version space, there are more complex selection methods suggested by [23], but it has been observed that the advantage of those are not significant when compared to their high computational costs.

Active Learning with Small Pools: The basic working principle of SVM active learning is: i) learn an SVM on the existing training data, ii) select the closest instance to the hyperplane, and iii) add the new selected instance to the training set and train again. In classical active learning [23], the search for the most informative instance is performed over the entire dataset. Note that, each iteration of active learning involves the recomputation of each training example's distance to the new hyperplane. Therefore, for large datasets, searching the entire training set is a very time consuming and computationally expensive task. We believe that we do not have to search the entire set at each iteration.

By using the "59 trick" [22], we propose a selection method, which does not necessitate a full search through the entire dataset but locates an approximate most informative sample by examining a small constant number of randomly chosen samples. The method picks L (L $\ll$ # training instances) random training samples in each iteration and selects the best (closest to the hyperplane) among them. Suppose, instead of picking the closest instance among all the training samples $X_N = (x_1, x_2, \cdots, x_N)$ at each iteration, we first pick a random subset X_L, $L \ll N$ and select the closest sample x_i from X_L based on the condition that x_i is among the top $p\%$ closest instances in X_N with probability $(1 - \eta)$. Any numerical modification to these constraints can be met by varying the size of L, and is independent of N. To demonstrate, the probability that at least one of the L instances is among the closest $p\%$ is $1-(1-p\%)^L$. Due to the requirement of $(1-\eta)$ probability, we have

$$1 - (1 - p\%)^L = 1 - \eta \tag{6}$$

which follows the solution of L in terms of η and p

$$L = \log \eta \; / \; \log(1 - p\%) \tag{7}$$

For example, the active learner will pick one instance, with 95% probability, that is among the top 5% closest instances to the hyperplane, by randomly sampling only $\lceil \log(.05)/\log(.95) \rceil = 59$ instances regardless of the training set size. This approach scales well since the size of the subset L is independent of the training set size N, requires significantly less training time and does not have an adverse effect on the classification performance of the learner.

In our experiments, we set $L = 59$ which means we pick 59 random instances to form the query pool at each learning step and pick the closest instance to the hyperplane from this pool. Figure 2 shows the comparisons of PRBEP and g-means performances of the proposed method AL(random pool) and the traditional active

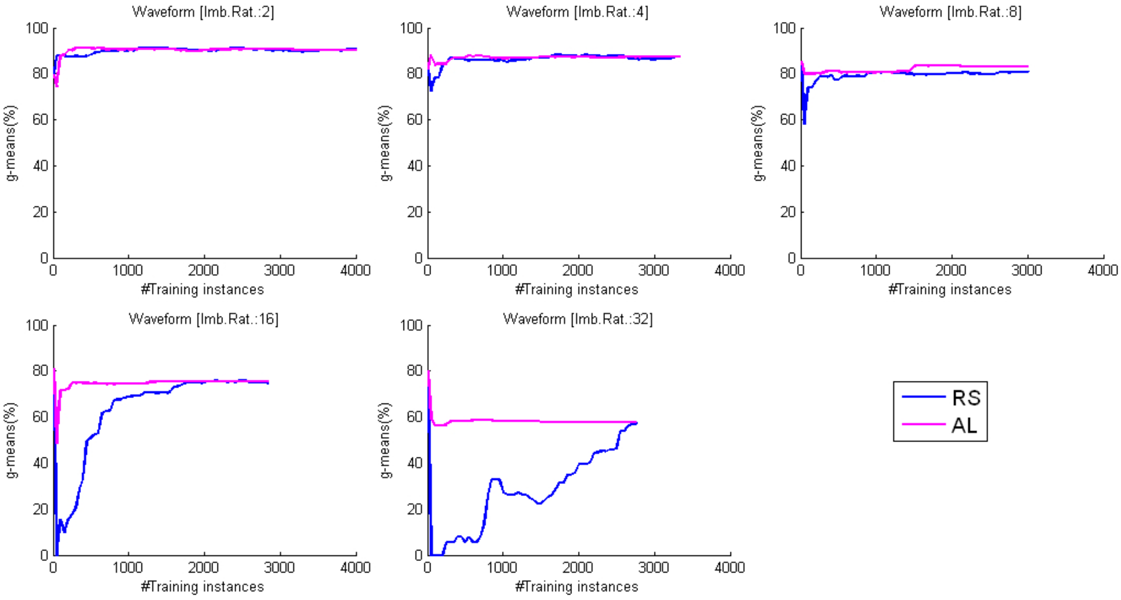

Figure 5: Comparison of g-means of AL and RS on the waveform datasets with different imbalance ratios (Imb.R.=2, 4, 8, 16, 32).

learning method AL(full search) [23]. RS corresponds to random sampling where instances are selected randomly. As Figure 2 depicts, the proposed active learning method with small pools achieves as good prediction performance as the traditional active learning method. Moreover, the proposed strategy is 4 to 10 times faster than the traditional active learning for the given datasets.

3.3 Online SVM for Active Learning

Online learning algorithms are usually associated with problems where the complete training set is not available. However, in cases where the complete training set is available, their computational properties can be leveraged for faster classification and incremental learning. In our framework, we use an online SVM algorithm, LASVM [3] instead of a traditional batch SVM tool (e.g., libsvm, SVM^{light}). LASVM is an online kernel classifier which relies on the traditional soft margin SVM formulation. LASVM yields the classification accuracy rates of the state-of-the art traditional SVM solvers but requires less computational resources. Traditional SVM works in a batch setting where all the training instances are used to form the one and final model. LASVM, on the other hand, works in an online setting, where its model is continually modified as it processes training instances one by one. Each LASVM iteration receives a fresh training example and tries to optimize the dual cost function in Equation (3) using feasible direction searches.

Online learning algorithms can select the new data to process either by random or active selection.They can integrate the information of the new seen data to the system without training all the samples again, hence they can incrementally build a learner. This working principle of LASVM leads to speed improvement and less memory demand which makes the algorithm applicable to very large datasets. More importantly, this incremental working principle suits the nature of active learning in a much better way than the batch algorithms. The new informative instance selected by active learning can be integrated to the existing model without retraining all the samples repeatedly. Empirical evidence indicates that a single presentation of each training example to the algorithm is sufficient to achieve training errors comparable to those achieved by the SVM solution [3]. In section 3.4 we also show that if we use an early stopping criteria in active sample selection, we do not have to introduce all the training instances to the learner.

3.4 Active Learning with Early Stopping

Early stopping criteria is advantageous to the active learning method since it converges to the solution faster than the random sample selection method. A theoretically sound method to stop training is when the examples in the margin are exhausted. To check if there are still unseen training instances in the margin, the distance of the new selected instance is compared to the support vectors of the current model. If the new selected instance by active learning (closest to the hyperplane) is not closer than any of the support vectors, we conclude that the margin is exhausted. A practical implementation of this idea is to count the number of support vectors during the active learning training process. If the number of the support vectors stabilizes, it implies that all possible support vectors have been selected by the active learning method.

In order to analyze this method, we conducted a 3-fold cross-validation on one of the datasets (see Figure 4). In cross-validation, $2/3$ of the training set is used for training and the remaining $1/3$ is reserved as the hold-out dataset. Since the training set distribution is representative of the test set distribution, we believe that the algorithm's behavior would most likely be the same in the test set. As can be seen in Figure 4, in active learning setups, after using certain number of labeled training data, the number of support vectors saturates and g-means levels off as well. Those graphs support the idea that the model does not change after the system observes enough informative samples. Further, adding more training data after this point does not make a remarkable change in the model and consequently in prediction performance. Notice that in Figure 4 the vertical line indicates the suggested early stopping point and it is approximately equal in all three folds. As a result, we adopt the early stopping strategy of examining the number of support vectors in the entire training datasets without performing cross-validation.

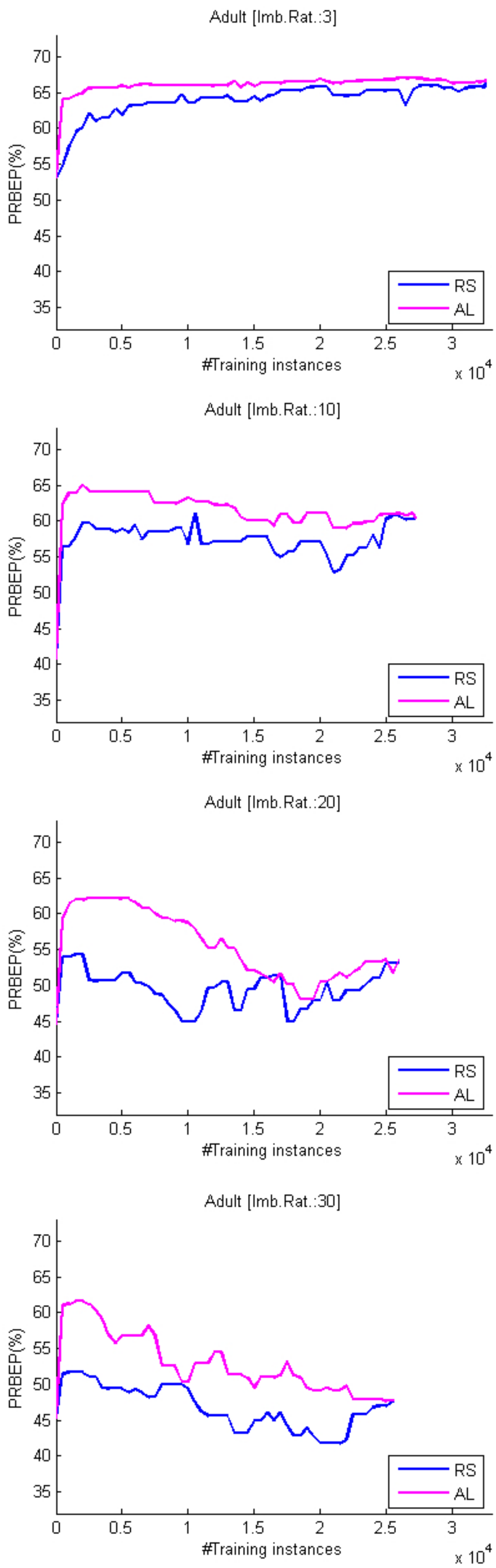

Figure 6: Comparison of PRBEP of AL and RS on the adult datasets with different imbalance ratios (Imb.R.=3, 10, 20, 30).

4. PERFORMANCE METRICS

Classification accuracy is not a good metric to evaluate classifiers in applications with class imbalance problem. SVMs have to achieve a tradeoff between maximizing the margin and minimizing the empirical error. In the non-separable case, if the misclassification penalty C is very small, SVM learner simply tends to classify every example as negative. This extreme approach makes the *margin* the largest while making no classification errors on the negative instances. The only error is the cumulative error of the positive instances which are already few in numbers. Considering an imbalance ratio of 99 to 1, a classifier that classifies everything as negative, will be 99% accurate but it will not have any practical use as it can not identify the positive instances.

For evaluation of our results, we use several other prediction performance metrics such as g-means, AUC and PRBEP which are commonly used in imbalanced data classification. g-means [14] is denoted as $g = \sqrt{sensitivity \cdot specifity}$ where sensitivity is the accuracy on the positive instances given as $TruePos./(TruePos. + FalseNeg.)$ and specificity is the accuracy on the negative instances given as $TrueNeg./(TrueNeg. + FalsePos.)$.

The Receiver Operating Curve (ROC) displays the relationship between sensitivity and specificity at all possible thresholds for a binary classification scoring model, when applied to independent test data. In other words, ROC curve is a plot of the true positive rate against the false positive rate as the decision threshold is changed. The *area under the ROC curve* (AUC) is a numerical measure of a model's discrimination performance and shows how successfully and correctly the model separates the positive and negative observations and ranks them. Since AUC metric evaluates the classifier across the entire range of decision thresholds, it gives a good overview about the performance when the operating condition for the classifier is unknown or the classifier is expected to be used in situations with significantly different class distributions.

Precision Recall Break-Even Point (PRBEP) is another commonly used performance metric for imbalanced data classification. PRBEP is the accuracy of the positive class at the threshold where precision equals to recall. Precision is defined as $TruePos./(TruePos. + FalsePos.)$ and recall is defined $TruePos./(TruePos.+FalseNeg.)$

5. DATASETS

We study the performance of the algorithm on various benchmark real-world datasets. The overview of the datasets are given in Table 2. The *Reuters-21578* is a popular text mining benchmark dataset. We test the algorithms with 8 of the top 10 most populated categories of *Reuters-21578*. We did not use categories 'earn' and 'acq' since their class imbalance ratios are not high enough. As a text dataset, we also used 5 categories from CiteSeer[1] data. We used 4 benchmark datasets from the popular UCI Machine Learning Repository as well. *Letter* and *satimage* are image datasets. The 'letter A' is used as the positive class in *letter* and 'class 4' (damp grey soil) is used as positive class in *satimage*. *Abalone* is a biology dataset. respectively. In *abalone*, instances labeled as 'class 7' are used to form the positive class. *MNIST* and *USPS* are OCR data of handwritten digits and 'digit 8' is used as a positive class in *Mnist*. *Adult* is a census dataset to predict if the income of a person is greater than 50K based on several census parameters, such as age, education, marital status etc. The training set consists of 32,562 instances and the class imbalance

[1] http://citeseer.ist.psu.edu

Table 1: Comparison of g-means and AUC for AL and RS with entire training data (Batch). Support vector ratios are given at the saturation point. Data efficiency corresponds to the percentage of training instances which AL processes to reach saturation.

	Dataset	g-means (%)		AUC (%)		Imb.	SV- /	Data
		Batch	AL	Batch	AL	Rat.	SV+	Efficiency
Reuters	Corn	85.55	86.59	99.95	99.95	41.9	3.13	11.6%
	Crude	88.34	89.51	99.74	99.74	19.0	2.64	22.6%
	Grain	91.56	91.56	99.91	99.91	16.9	3.08	29.6%
	Interest	78.45	78.46	99.01	99.04	21.4	2.19	30.9%
	Money-fx	81.43	82.79	98.69	98.71	13.4	2.19	18.7%
	Ship	75.66	74.92	99.79	99.80	38.4	4.28	20.6%
	Trade	82.52	82.52	99.23	99.26	20.1	2.22	15.4%
	Wheat	89.54	89.55	99.64	99.69	35.7	3.38	11.6%
CiteSeer	AI	87.83	88.58	94.82	94.69	4.3	1.85	33.4%
	COMM	93.02	93.65	98.13	98.18	4.2	2.47	21.3%
	CRYPT	98.75	98.87	99.95	99.95	11.0	2.58	15.2%
	DB	92.39	92.39	98.28	98.46	7.1	2.50	18.2%
	OS	91.95	92.03	98.27	98.20	24.2	3.52	36.1%
UCI	Abalone-7	100.0	100.0	100.0	100.0	9.7	1.38	24.0%
	Letter-A	99.28	99.54	99.99	99.99	24.4	1.46	27.8%
	Satimage	82.41	83.30	95.13	95.75	9.7	2.62	41.7%
	USPS	99.22	99.25	99.98	99.98	4.9	1.50	6.8%
	MNIST-8	98.47	98.37	99.97	99.97	9.3	1.59	11.7%

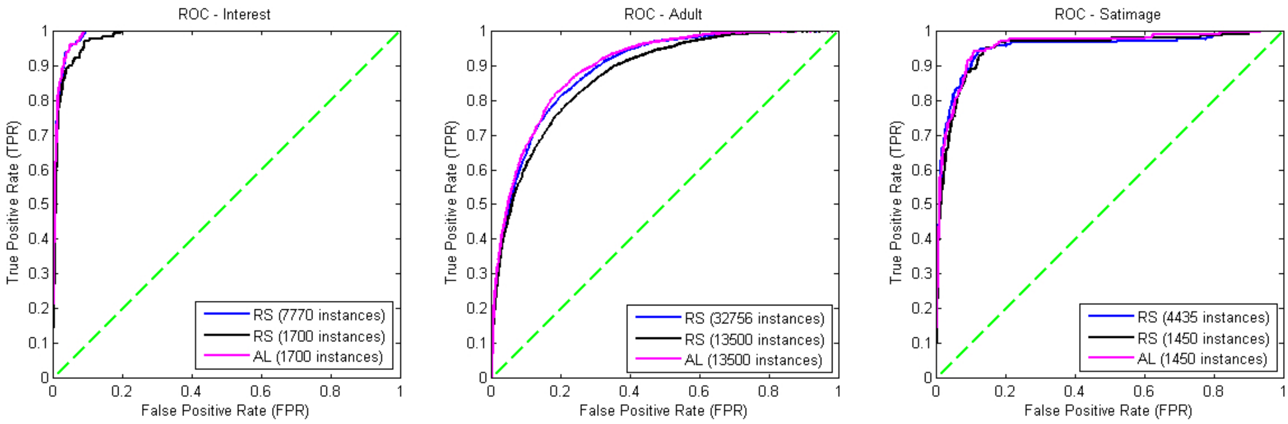

Figure 7: Comparison of ROC curves of AL, RS (early stopped at the same number of instances as AL) and RS (with all training data) in Interest, Adult and Satimage datasets.

ratio is 3. *Waveform* is a popular artificial dataset used commonly in simulation studies. These datasets cover a wide range of data imbalance ratio.

6. EXPERIMENTS AND EMPIRICAL EVALUATION

We first conduct experiments to compare the performance of the proposed active learning strategy AL(random pool) with the traditional active learning method, AL(full search). The results show that with the proposed method, we can make faster active learning without sacrificing any prediction performance (see Figure 2). In the rest of the paper, we refer to our proposed method as AL since it is the only active learning method that we used afterwards.

In order to make a thorough analysis on the effect of AL to imbalanced data classification, we examine its performance by varying class imbalance ratios using two performance metrics. We randomly remove the instances from the minority class in *Waveform* and *Adult* datasets to achieve different data imbalance ratios. Comparisons of g-means of AL and RS in Figure 5 show that the prediction performance of AL is less sensitive to the class imbalance ratio changes than that of the RS. Comparisons of another performance metric PRBEP in Figure 6 give even more interesting results. As the class imbalance ratio is increased, AL curves display peaks in the early steps of the learning. This implies that by using an early stopping criteria AL can give higher prediction performance than RS can possibly achieve even after using all the training data. Figure 6 curves allow us to think that addition of any instances to the learning model after finding the informative instances can be detrimental to the prediction performance of the classifier. This finding strengthens the idea of applying an early stopping to the active learning algorithms.

We also compared the performance of early stopped AL with Batch algorithm. Table 1 presents the g-means and the AUC values of the two methods. Data efficiency column for AL indicates that by processing only a portion of the instances from the training set, AL can achieve similar or even higher prediction performance than that of Batch which sees all the training instances. Another important observation from Table 1 is that support vector imbalance ratios in the final models are much less than the class imbalance ratios of the datasets. This confirms our discussion of Figure 3 in section 3. The class imbalance ratio within the margins are much less than the class imbalance ratio of the entire data and active learning can be used to reach those informative instances which most likely become support vectors without seeing all the training instances.

In order to evaluate the methods at different thresholds, we also investigate the ROC curves as given in Figure 7. The ROC curves of AL are similar and sometimes better than of the Batch algorithm (RS, seeing all the training instances). The AUC of AL and Batch are 0.8980 and 0.8910 respectively in the *Adult* dataset. At the same number of training instances where AL is early stopped, AUC of RS can be substantially lower. As Figure 7 shows, the ROC curve of AL is markedly higher than that of RS (early stopping) and the AUC values are 0.8980 and 0.8725 respectively for *Adult* dataset. These results suggest that AL converges faster than RS using fewer and informative instances and AL can get even higher prediction performance than the Batch algorithm by processing only a portion of the training set.

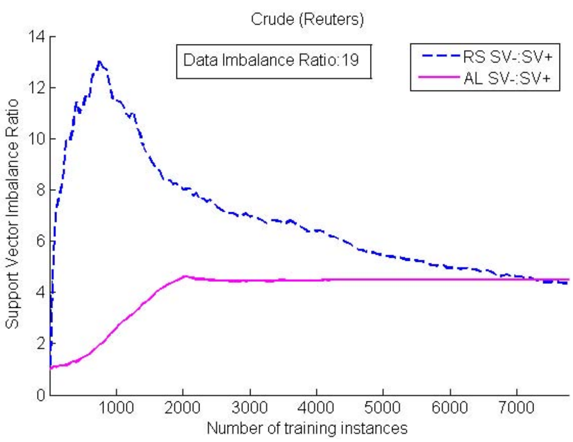

Figure 8: Support Vector ratios in AL and RS

In Figure 8, we investigate how the number of support vectors changes in AL and Random Sampling (RS). With random sampling, the instances are selected for the learner randomly from the entire pool of the training data. Therefore, the support vector imbalance ratio quickly approaches the data imbalance ratio. As learning continues, the learner should gradually see all the instances within the final margin and the support vector imbalance ratio decreases. When RS finishes learning, the support vector imbalance ratio is the data imbalance ratio within the margin. The support vector imbalance ratio curve of AL is drastically different than RS. AL intelligently picks the instances closest to the margin in each step. Since the data imbalance ratio within the margin is lower than data imbalance ratio, the support vectors in AL are more balanced than RS during learning. Using AL, the model saturates by seeing only 2000 (among 7770) training instances and reaches the final support vector imbalance ratio. Note that both methods achieve similar support vector imbalance ratios when learning finishes, but AL achieves this in the early steps of the learning.

Table 2: Overview of the datasets.

Dataset		#Feat.	#Pos	#Neg	Ratio	c	γ
Reuters	Crude	8315	389	7381	19.0	2	1
	Grain	8315	433	7337	16.9	2	1
	Interest	8315	347	7423	21.4	1	2
	Money-fx	8315	538	7232	13.4	1	0.5
	Ship	8315	197	7573	38.4	1	0.5
	Wheat	8315	212	7558	35.7	1	0.5
CiteSeer	AI	6946	1420	5353	4.3	50	0.1
	COMM	6946	1252	5341	4.2	50	0.1
	Crypt	6946	552	6041	11.0	50	0.1
	DB	6946	819	5775	7.1	50	0.1
	OS	6946	262	6331	24.2	50	0.1
UCI	Abalone-7	9	352	3407	9.7	100	0.01
	Letter-A	16	710	17290	24.4	10	0.01
	Satimage	36	415	4020	9.69	50	0.001
USPS		256	1232	6097	5.0	1000	2
MNIST-8		780	5851	54149	9.3	1000	0.02

We compare the AL method discussed in this paper with several other strategies as well. Among them, undersampling (US), and an oversampling method (SMOTE) are examples of resampling techniques which require preprocessing. Recent research showed that oversampling at random does not help to improve prediction performance [12] therefore we use a more complex oversampling method (SMOTE). As an algorithmic method to compare, we use the method of assigning different costs (DC) to the positive and negative classes as the misclassification penalty parameter. For instance, if the imbalance ratio of the data is 19:1 in favor of the negative class, the cost of misclassifying a positive instance is set to be 19 times greater than that of misclassifying a negative one. We use LASVM[2], an online SVM tool, in all experiments. Other than the results of the methods addressing class imbalance problem, we also give results of Batch algorithm with the original training set to form a baseline. LASVM is run in random sampling mode for US, SMOTE and DC.

We give the comparisons of the methods for g-means performance metric for several datasets in Figure 9. The right border of the shaded pink area is the place where the aforementioned early stopping strategy is applied. The curves in the graphs are averages of 10 runs. For completeness we did not stop the AL experiments at the early stopping point but allow them to run on the entire training set. We present the PRBEP of the methods and the total running times of the SMOTE and AL on 18 benchmark and real-world datasets in Table 3. The results for active learning in Table 3 depict the results in the early stopping points. The results for the other methods in Table 3 depict the values at the end of the curves –when trained with the entire dataset– since those methods do not employ any early stopping criteria. We did not apply early stopping criteria to the other methods because as observed from Figure 9, no early stopping criteria would achieve a comparable training time with of AL's training time without a significant loss in their prediction performance based on convergence time. The other methods converge to similar levels of g-means when nearly all training instances are used, and applying an early stopping criteria would have little, if any, effect on their training times.

Since AL involves discarding some instances from the training set, it can be perceived as a type of undersampling method. Unlike US which discards majority samples randomly, AL performs an intelligent search for the most informative ones adaptively in each

[2] Available at http://leon.bottou.org/projects/lasvm

Table 3: **Comparison of PRBEP and training time.**

Metric		PRBEP					Training time (sec.)	
Dataset		Batch	US	SMOTE	DC	AL	SMOTE	AL
Reuters	Corn	91.07	78.57	91.07	89.28	89.29	87	16
	Crude	87.83	85.70	87.83	87.83	87.83	129	41
	Grain	92.62	89.93	91.44	91.94	91.94	205	50
	Interest	76.33	74.04	77.86	75.57	75.57	116	42
	Money-fx	73.74	74.30	75.42	75.42	76.54	331	35
	Ship	86.52	86.50	88.76	89.89	89.89	49	32
	Trade	77.77	76.92	77.77	77.78	78.63	215	38
	Wheat	84.51	81.61	84.51	84.51	85.92	54	25
CiteSeer	AI	78.80	80.68	78.99	78.79	79.17	1402	125
	COMM	86.59	86.76	86.59	86.59	86.77	1707	75
	CRYPT	97.89	97.47	97.89	97.89	97.89	310	19
	DB	86.36	86.61	86.98	86.36	86.36	526	41
	OS	84.07	83.19	84.07	84.07	84.07	93	23
UCI	Abalone-7	100.0	100.0	100.0	100.0	100.0	16	4
	Letter-A	99.48	96.45	99.24	99.35	99.35	86	3
	Satimage	73.46	68.72	73.46	73.93	73.93	63	21
USPS		98.44	98.44	98.13	98.44	98.75	4328	13
MNIST-8		97.63	97.02	97.74	97.63	97.74	83,339	1,048

iteration according to the current hyperplane. In datasets where class imbalance ratio is high such as *corn*, *wheat*, *letter* and *satimage*, we observe significant decrease in PRBEP of US (see Table 3). Note that US's undersampling rate for the majority class in each category is set to the same value as the final support vector ratio which AL reaches in the early stopping point and RS reaches when it sees the entire training data. Although the class imbalance ratio provided to the learner in AL and US are the same, AL achieves significantly better PRBEP performance metric than US. The Wilcoxon signed-rank test (2-tailed) reveals that the zero median hypothesis can be rejected at the significance level 1% (p=0.0015), implying that AL performs statistically better than US in these 18 datasets. These results reveal the importance of using the informative instances for learning.

Table 4: Comparison of ranks of different methods in PRBEP. The values in bold correspond to the cases where AL win. AL wins in 12 out of 18 cases in PRBEP.

Metric		Rank				
Dataset		Batch	US	SMOTE	DC	AL
Reuters	Corn	1	5	1	4	3
	Crude	1	5	1	1	**1**
	Grain	1	5	4	2	2
	Interest	2	5	1	3	3
	Money-fx	5	4	2	2	**1**
	Ship	4	5	3	1	**1**
	Trade	3	5	3	2	**1**
	Wheat	2	5	2	2	**1**
CiteSeer	AI	4	1	3	5	2
	COMM	3	2	3	3	**1**
	CRYPT	1	5	1	1	**1**
	DB	3	2	1	3	3
	OS	1	5	1	1	**1**
UCI	Abalone-7	1	1	1	1	**1**
	Letter-A	1	5	4	2	2
	Satimage	3	5	3	1	**1**
USPS		2	2	5	2	**1**
MNIST-8		3	5	1	3	**1**
Avg. Rank		2.28	4.00	2.22	2.17	**1.50**

Table 4 presents the rank of PRBEP prediction performance of the five approaches in a variety of datasets. The values in bold correspond to the cases where AL wins and it's clear that winning cases are very frequent for AL (12 out of 18 cases). The average rank also indicates that AL achieves the best PRBEP among the five methods. SMOTE and DC achieve higher PRBEP than the Batch algorithm. The loss of information when undersampling the majority class affects US's prediction performance. Table 3 also gives the comparison of the computation times of the AL and SMOTE. Note that SMOTE requires significantly long preprocessing time which dominates the training time in large datasets, e.g., MNIST-8 dataset. The low computation cost, scalability and high prediction performance of AL suggest that AL can efficiently handle the class imbalance problem.

7. CONCLUSIONS

The class imbalance problem has been known to impact the prediction performance of classification algorithms. The results of this paper offer a better understanding of the effect of the active learning on imbalanced datasets. We first propose an efficient active learning method which selects informative instances from a randomly picked small pool of examples rather than making a full search in the entire training set. This strategy renders active learning to be applicable to very large datasets which otherwise would be computationally very expensive. Combined with the early stopping heuristics, active learning achieves a fast and scalable solution without sacrificing prediction performance. We then show that the proposed active learning strategy can be used to address the class imbalance problem. In simulation studies, we demonstrate that as the imbalance ratio increases, active learning can achieve better prediction performance than random sampling by only using the informative portion of the training set. By focusing the learning on the instances around the classification boundary, more balanced class distributions can be provided to the learner in the earlier steps of the learning. Our empirical results on a variety of real-world datasets allow us to conclude that active learning is comparable or even better than other popular resampling methods in dealing with imbalanced data classification.

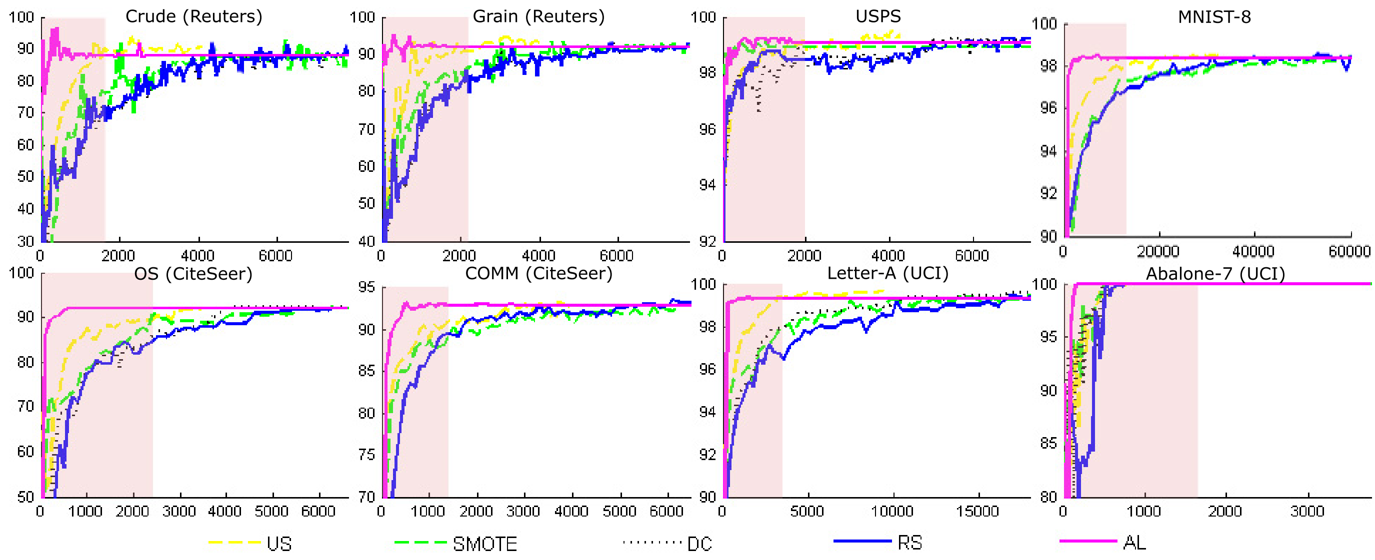

Figure 9: Comparisons of g-means. The right border of the shaded area corresponds to the early stopping point.

8. REFERENCES

[1] N. Abe. Invited talk: Sampling approaches to learning from imbalanced datasets: Active learning, cost sensitive learning and beyond. *Proc. of ICML Workshop: Learning from Imbalanced Data Sets*, 2003.

[2] R. Akbani, S. Kwek, and N. Japkowicz. Applying support vector machines to imbalanced datasets. *Proc. of European Conference on Machine Learning*, pages 39–50, 2004.

[3] A. Bordes, S. Ertekin, J. Weston, and L. Bottou. Fast kernel classifiers with online and active learning. *Journal of Machine Learning Research (JMLR)*, 6:1579–1619, 2005.

[4] P. K. Chan and S. J. Stolfo. Toward scalable learning with non-uniform class and cost distributions: A case study in credit card fraud detection. In *Proc. of ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, 1998.

[5] N. V. Chawla, K. W. Bowyer., L. O. Hall, and W. P. Kegelmeyer. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research (JAIR)*, 16:321–357, 2002.

[6] P. Domingos. Metacost: A general method for making classifiers cost-sensitive. In *Proc. of the ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, 1999.

[7] S. Dumais, J. Platt, D. Heckerman, and M. Sahami. Inductive learning algorithms and representations for text categorization. In *Proc. of Int. Conference on Information and Knowledge Management (CIKM)*, 1998.

[8] J. W. Grzymala-Busse, Z. Zheng, L. K. Goodwin, and W. J. Grzymala-Busse. An approach to imbalanced datasets based on changing rule strength. In *Proc. of In Learning from Imbalanced Datasets, AAAI Workshop*, 2000.

[9] J. Huang, S. Ertekin, and C. L. Giles. Efficient name disambiguation for large scale datasets. In *Proc. of European Conference on Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD)*, 2006.

[10] N. Japkowicz. A novelty detection approach to classification. In *Proc. of the Int. Joint Conference on Artificial Intelligence (IJCAI)*, pages 518–523, 1995.

[11] N. Japkowicz. The class imbalance problem: Significance and strategies. In *Proc. of 2000 Int. Conference on Artificial Intelligence (IC-AI'2000)*, volume 1, pages 111–117, 2000.

[12] N. Japkowicz and S. Stephen. The class imbalance problem: A systematic study. *Intelligent Data Analysis*, 2002.

[13] M. Kubat, R. C. Holte, and S. Matwin. Machine learning for the detection of oil spills in satellite radar images. *Machine Learning*, 30(2-3):195–215, 1998.

[14] M. Kubat and S. Matwin. Addressing the curse of imbalanced training datasets: One sided selection. *Proc. of Int. Conference on Machine Learning (ICML)*, 30(2-3), 1997.

[15] C. X. Ling and C. Li. Data mining for direct marketing: Problems and solutions. In *Knowledge Discovery and Data Mining*, pages 73–79, 1998.

[16] X.-Y. Liu, J. Wu, and Z.-H. Zhou. Exploratory under-sampling for class-imbalance learning. In *Proc. of the International Conference on Data Mining (ICDM)*, 2006.

[17] M. Pazzani, C. Merz, P. Murphy, K. Ali, T. Hume, and C. Brunk. Reducing misclassification costs. In *Proc. of 11th Int. Conference on Machine Learning (ICML)*, 1994.

[18] F. Provost. Machine learning from imbalanced datasets 101. In *Proc. of AAAI Workshop on Imbalanced Data Sets*, 2000.

[19] P. Radivojac, N. V. Chawla, A. K. Dunker, and Z. Obradovic. Classification and knowledge discovery in protein databases. *Journal of Biomedical Informatics*, 37(4):224–239, 2004.

[20] B. Raskutti and A. Kowalczyk. Extreme re-balancing for svms: a case study. *SIGKDD Explorations Newsletter*, 6(1):60–69, 2004.

[21] G. Schohn and D. Cohn. Less is more: Active learning with support vector machines. In *Proc. of the 17th Int. Conference on Machine Learning (ICML)*, pages 839–846, 2000.

[22] A. J. Smola and B. Schölkopf. Sparse greedy matrix approximation for machine learning. In *Proc. of 17th Int. Conference on Machine Learning (ICML)*.

[23] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research (JMLR)*, 2:45–66, 2002.

[24] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, 1995.