Active Learning for Class Imbalance Problem

Seyda Ertekin
Department of Computer Science and Engineering
Pennsylvania State University
University Park, PA 16802
sertekin@cse.psu.edu

Jian Huang
College of Information Sciences and Technology
Pennsylvania State University
University Park, PA 16802
jhuang@ist.psu.edu

C. Lee Giles
College of Information Sciences and Technology
Pennsylvania State University
University Park, PA 16802
giles@ist.psu.edu

ABSTRACT

The class imbalance problem has been known to hinder the learning performance of classification algorithms. Various real-world classification tasks such as text categorization suffer from this phenomenon. We demonstrate that active learning is capable of solving the problem.

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

A dataset is called imbalanced if at least one of the classes are represented by significantly less number of instances than the others. In imbalanced data classification, the class boundary learned by the standard machine learning algorithms can be severely skewed toward the positive class. Thus, the false-negative rate can be excessively high. One major research direction to overcome the class imbalance problem is to resample the original training dataset, either by *oversampling* the minority class and/or *undersampling* the majority class until the classes are represented in a more balanced way. Undersampling may discard useful data that could be important for the learning process. Oversampling causes longer training time and inefficiency in terms of memory due to the increased number of training instances and it suffers from high computational costs for preprocessing the data. The underlying motivation for resampling methods is to provide the learner with a training set having more balanced classes. We show that Active Learning (AL) strategy can be a more efficient alternative to resampling methods to form a balanced training set for the learner in early stages of the learning. We also propose an efficient Support Vector Machine (SVM) active learning strategy which queries a small pool of data at each iterative step instead of querying the entire dataset. We present that active learning with early stopping can achieve a faster and scalable solution without sacrificing prediction performance.

Amsterdam, Netherlands SIGIR 2007

2. EFFICIENT ACTIVE LEARNING

The basic SVM based active learning selects the closest instance to the current hyperplane from the unseen training data and adds it to the training set to retrain the model. In classical active learning [3], the search for the most informative (closest) instance is done through the entire unseen dataset. Each iteration of active learning involves the recomputation of the distances of each instance to the new hyperplane. Thus, for large datasets, searching the entire training set is very time-consuming and computationally expensive.

We propose a selection method which will 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

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

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

$$L = \log\eta \;/\; \log(1-p\%) \tag{2}$$

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 $L = 59$.

Early Stopping: In SVM learning the classification boundary (hyperplane) is only determined by support vectors. This means that there is no point of adding new instances to the model after the

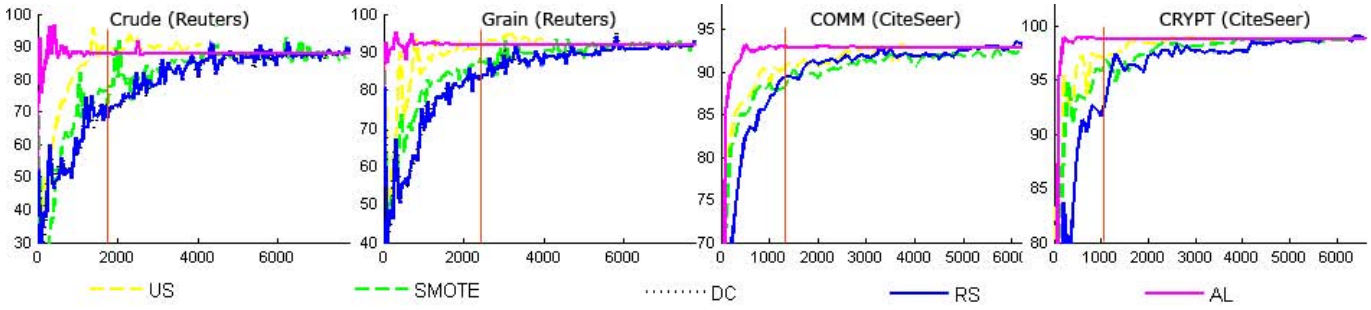

Figure 1: Comparisons of g-means. The vertical line corresponds to the early stopping point.

Table 1: The running time of SMOTE is composed of the preprocessing time and the training time. There are 7,770 and 6,593 training instances in Reuters and CiteSeer respectively.

	Dataset	Train time (sec)		g-means		Imb.	SV- /	Data
		SMOTE	AL	RS	AL	Rat.	SV+	Eff.(%)
Reuters	Crude	123	55	88.3	91.2	19.0	2.64	22.6%
	Grain	128	52	91.5	91.9	16.9	3.08	29.6%
	Interest	84	33	78.4	78.9	21.4	2.19	30.9%
	Money-fx	161	57	81.4	81.1	13.4	2.19	18.7%
	Ship	48	48	75.6	74.9	38.4	4.28	20.6%
	Wheat	49	37	89.5	89.5	35.7	3.38	11.6%
CiteSeer	AI	215	73	88.0	87.8	4.3	1.83	40.9%
	COMM	242	50	93.0	92.8	4.2	2.47	21.3%
	CRYPT	74	20	98.7	98.9	11.0	2.58	15.2%
	DB	150	40	92.3	92.4	7.1	2.21	27.3%
	OS	37	21	91.9	91.0	24.2	3.52	36.1%

number of support vectors saturates. 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 and the rest of the training instances are redundant. Therefore, we choose our stopping point where the number of support vectors saturates.

3. EXPERIMENTS

We compare the active learning method (AL) with several other strategies. Among them undersampling (US) and oversampling method (SMOTE [2]) are examples of resampling techniques. DC is the method which uses different costs (higher for the positive class) for misclassification penalties. As a classification algorithm, we used LASVM [1], an online SVM tool for all the experiments. Online SVMs suits the nature of active sample selection strategy better than the batch SVMs due to their incremental learning steps. We also show random sample selection of LASVM (RS) on the original training set to form the baseline. LASVM is also run in random sample selection mode with US, SMOTE and DC. We conducted the experiments on two text datasets: Reuters-21578 and CiteSeer. Due to space constraints, we only show results for some categories of those datasets.

We use the g-means metric which is a common practice in the performance evaluation of algorithms in imbalanced data classification. g-means is denoted as $g = \sqrt{sensitivity \cdot specifity}$ where sensitivity and specificity are the accuracies on the positive and negative instances respectively.

Figure 1 depicts the g-means performance of the different methods. For completeness we did not cut the active learning experiments at the early stopping point but allowed them to run on the entire training set. AL g-means curves saturate after using small portion of the training data. Those graphs support the idea that adding more training data after seeing the informative samples does not remarkably change the model and consequently the prediction performance. The values of AL curves on the vertical lines in Figure 1 show the g-means of AL at the early stopping points.

The training times for AL in Table 1 show the time till the early stopping point. The results for the other methods present the training times at the end of the curves. We did not apply early stopping criteria to the other methods because as observed from Figure 1 the other methods converge to similar levels of g-means when nearly all training instances are used. Thus, no early stopping criteria would achieve a comparable training time with that of ALs without a significant loss in their prediction performance based on convergence time. We also give the imbalance ratios of the positive and negative classes in the training sets for each category, and the imbalance ratios of the negative and positive support vectors in the final models. Those more balanced models are achieved in much earlier steps of the learning in AL while the other methods have to see the entire training dataset. Data efficiency of AL in Table 1 shows what percentage of the training instances are used in the model to achieve the balanced model. Since other methods use all the training instances, data efficiency is not applicable for those.

4. CONCLUSION

Experimental results on text datasets show that our method can achieve a significant decrease in the training time, while maintaining the same or achieving even higher g-means values by using less number of training instances in the SVM model. The efficient method for active selection of informative instances from a randomly picked small pool of samples removes the need for making a search through the entire dataset. This strategy makes active learning scalable to large datasets. Combined with the early stopping heuristics, active learning can be an alternative method for solving the class imbalance problem.

5. REFERENCES

[1] 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, September 2005.

[2] N. V. Chawla, K. W. Bowyer., L. O. Hall, and W. P. Kegelmeyer. Smote: Synthetic minority over-sampling technique. *J. Artificial Intelligence Research*, vol. 16, 2002.

[3] 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.