JMLR: Workshop and Conference Proceedings 16 (2011) 85–98 Workshop on Active Learning and Experimental Design

Stochastic Semi-supervised Learning on Partially Labeled Imbalanced Data

Jianjun Xie JIANJUNXIE@GMAIL.COM
CoreLogic, 703 Palomar Airport Road, Carlsbad, CA 92021, USA

Tao Xiong TAO.XIONG@GMAIL.COM
eBay Inc., 2145 Hamilton Avenue, San Jose, CA 95125, USA

Editor: I. Guyon, G. Cawley, G. Dror, V. Lemaire, and A. Statnikov

Abstract

In this paper, we describe the stochastic semi-supervised learning approach that we used in our submission to all six tasks in 2009-2010 Active Learning Challenge. The method is designed to tackle the binary classification problem under the condition that the number of labeled data points is extremely small and the two classes are highly imbalanced. It starts with only one positive seed given by the contest organizer. We randomly pick additional unlabeled data points and treat them as "negative" seeds based on the fact that the positive label is rare across all datasets. A classifier is trained using the "labeled" data points and then is used to predict the unlabeled dataset. We take the final result to be the average of n stochastic iterations. Supervised learning was used as a large number of labels were purchased. Our approach is shown to work well in 5 out of 6 datasets. The overall results ranked 3rd in the contest.

Keywords: Active Learning, Semi-supervised Learning, Gradient Boosting Decision Tree

1. Introduction

The 2009-2010 active learning challenge consisted of six real world datasets from six different domains: handwriting recognition (A), marketing (B), chemo-informatics (C), text processing (D), embryology (E) and ecology (F) (Guyon et al., this volume). Each data set has a different number of features, a different number of records and a different positive label percentage. They are all binary classification problems with an imbalanced distribution of the two classes. Each dataset has been split into training and testing randomly. An initial positive seed is given in the training set. The participants were asked to submit the prediction to all the samples with unknown labels based on the queries had been made.

The prediction performance metric is the Area under the Learning Curve (ALC) which is referred to as the global score. A learning curve plots the Area Under the ROC curve (AUC) computed on all the samples with unknown labels, as a function of the number of queried labels (including the initial seed). In order to emphasize the model performance with few known labels, the x-axis is $\log 2$ scaled.

Six development datasets were made available before the final contest datasets were released. This provided the participants the opportunity to develop query strategies as well as to select the best learning method. Even though the final datasets are extracted from the same domain as the development datasets, they are different enough so that participants are not able to directly apply what they learned from development datasets. The domain of each final dataset and the label distribution are anonymized.

Active learning algorithms have seen many applications during the past decade in such areas as text classification (Tong and Koller, 2001), image classification (Luo et al., 2005), software testing (Bowring et al., 2004) and so on. Generally speaking, there are three learning scenarios of active learners: (i) membership query synthesis, (ii) stream-based selective sampling, and (iii) pool-based sampling. Pool-based active learning is the setting used in this challenge. In pool-based active learning, there are typically three strategies of querying unlabeled instances (Settles, 2009). First is uncertainty sampling. The examples whose predicted label (based on the current classifier estimate) is most ambiguous are picked first for label inquiry. Among others, measures of uncertainty include disagreement among oracles in Query by Committee (Freund et al., 1997), confidence of classification (Lewis and Gale, 1994), and distance to a decision boundary in SVMs (Tong and Koller, 2001). Second is called reducing future error, Roy and McCallum (2001) proposed to pick examples that minimize the generalization error probability. Because it is impossible to know future generalization errors, it uses the current classifier to estimate the probabilities for each unlabeled example. The third type uses ensembles of active learners. Baram et al. (2004) developed a master algorithm that picks the best expert from an ensemble of active learners depending on their performance. A more comprehensive and detailed literature review of those strategies can be found in (Settles, 2009).

When classifiers are trained, active learning algorithm usually takes advantage of the fact that both labeled and unlabeled data instances are available. Typically some form of semi-supervised learning algorithm is used for better performance. Examples of semi-supervised learning techniques include co-training (Blum and Mitchell, 1998), self-training (Rosenberg et al., 2005), cluster-and-label (Demiriz et al., 1999; Dara et al., 2002)and so on. A detailed literature survey on semi-supervised learning can be found in (Zhu, 2005).

In this contest, we proposed a stochastic semi-supervised learning approach to handle the active learning challenge when the number of labeled data is extremely small. We borrowed the concept of self-training in our logistic regression approach and the idea of cluster-and-label in our k-means clustering approach. We incorporated these ideas into a stochastic sample-train-label process. The details of the approach will be given in Section 2. We summarize our results and the comparison with others in Section 3. Finally, conclusion of our work is given in Section 4.

2. Our Approach

The method we used is a stochastic semi-supervised learning process. It was proposed mainly based on the following two facts in this contest. First, the number of available labeled examples is extremely small, while the number of unlabeled examples is abundant. Second, the positively labeled exemplars are rare comparing with negatively labeled ones. In other words, the probability of getting a negatively labeled exemplar through random sampling from the unlabeled pool is much higher than the probability of getting a positively labeled exemplar.

Two classifiers are used in the stochastic semi-supervised learning process. One is clustering and another is logistic regression. The criteria we used to choose clustering or logistic regression are based on the following factors.

1. Number of features. If the number of feature is very large, say more than 800, clustering is preferred. This is because clustering is an unsupervised approach. We can use the whole dataset to do clustering with the initially available label as seed. While logistic regression is a supervised learning, we have to get enough labels at the beginning to build a meaningful model. A large number of features will increase the complexity of the model with very limited available examples. Since dataset C and D are two datasets with the largest number of features (i.e. 12000 for C and 851 for D), we choose clustering for these two datasets.

2. Distribution of two classes. We prefer to use logistic regression if the two classes are extremely unbalanced. This is because if one classes is extremely rare, clustering method may treat them as outliers and ignore them. On the other hand, it is a lot easier to get right by randomly picking some unlabeled examples and labeling them as majority class. If the two classes are more balanced, we would choose clustering because it is too easy to be wrong for the initial labeling by random guess. Even though the class distribution was not available in the test dataset, it was not hard to figure out which development dataset it was corresponding to. We therefore chose clustering for dataset A since it corresponds to handwriting recognition which has the highest percentage of positive label in development dataset (37%). We decided to use logistic regression approach for the remaining 3 datasets.

Algorithm 1 details the steps we used in the contest before we did any label purchase. Each dataset is given one positive seed by the organizer to start with. For dataset A, C and D, we randomly pick another data point from the unlabeled data pool as a negative seed. We use these two seeds as our initial cluster centers for k-means clustering. We repeat this process n times, each time with a different randomly picked negative seed and the same positive seed. We label the cluster where the positive seed resides as positive cluster, the other one as negative cluster. We calculate the count of positive cluster membership of each data point after n iterations and use the normalized membership count as the prediction score.

For dataset B, E and F, we randomly pick 20 unlabeled data points as negative label for each positive label. This assumes that the positive label in these datasets was less than 5%. This is true in the corresponding development datasets. We build a logistic regression model using the "labeled" data points. We repeat n iterations of the above random sampling/modeling process and take the average score as the prediction score. This score is used to label all other unlabeled data (the concept of self-training). Final logistic regression model is built on dataset with the "derived" labels.

When more labels are available through the query, we mix these labeled data with the sampled "negative" data together as initial seeds (for k-means) or modeling dataset (for logistic regression). We repeat the stochastic process after each label query. The known labels are always kept in the modeling dataset, while the "derived" labels are changing each time. We put more weights on the labeled data points this way. The random sampling

Algorithm 1: Stochastic semi-supervised learning process

Given one positive label, N unlabeled examples:

```
1. For dataset A, C and D
    2. Set $i = 1$
        3. Randomly pick one example from $N$ unlabeled examples as "negative" example
        4. Use the positive label and the "negative" label as initial seeds, do $k$-means clustering on whole dataset with number of cluster = 2
        5. Label cluster where positive seed sits as positive, another one as negative
        6. Save cluster membership of each example $f_i(c)$ where $f(c = positive) = 1; f(c = negative) = 0$.
        7. Increase $i$ by 1. If $i < 100$ return to step 3
    8. Calculate final predicted score for each example using $\frac{1}{M}\sum_{i=1}^{M} f_i(c)$ where $M = 100$.
9. For dataset B, E and F
    10. Set $i = 1$
        11. Randomly pick 20 examples from $N$ unlabeled examples as "negative" examples
        12. Use the one positive label and the 20 "negative" labels as training set, build a logistic regression model
        13. Score the whole dataset, save score for each example $f_i$
        14. Increase $i$ by 1. If $i < 100$ return to step 11.
    15. Calculate average score for each example using $\frac{1}{M}\sum_{i=1}^{M} f_i$ where $M = 100$.
    16. Label highest 1% of score as positive examples, lowest 1% of score as negative examples, rebuild the logistic regression model (self-training).
    17. Calculate final score for each example using above logistic regression model.
```

process is repeated n time as described above. The final score is the arithmetic average of the n stochastic process. We take $n = 100$ in this contest.

The above semi-supervised learning approach is used when the number of available labels is extremely small. When the amount of the labeled data becomes large, we tend to use Gradient Boosting Decision Tree (TreeNet) (Friedman, 1999) as our classifier to generate prediction score. We use a switching threshold of approximately 200 in this work. This corresponds to the middle range of x value for all 6 datasets in the area under the learning curve plot because of the $\log 2$ scaling on the number of purchased labels. It is not possible to conduct an experiment to figure out the best threshold value in the contest since there is only one chance for each team to develop their approach. We heuristically obtained this value from our experiments on the development datasets. However, the test datasets were modified by the organizer so that they were different enough from the development sets even for the ones from the same domain. We took this value as a reference in the contest. For most of our label queries, we directly jumped to a large purchase (> 1000 labels) from a very small purchase (less than 100). We only built a supervised learning model on a single dataset (dataset A) with 233 purchased labels.

2.1. Dataset A: Handwriting Recognition

Every team had three chances to work on the datasets from this domain: development phase, contest phase and verification phase. Only the verification phase counted in the competition because every team got different labels during the contest stage for the same dataset, therefore the results were not comparable. This design was used to detect potential cheaters.

We will present the details of our experiments in this paper only on the contest phase and the verification phase. For these two phases, all the input fields are exactly same. The difference is that the training labels, as shown in Table 1 are altered. We can see that 688 negative training labels in verification phase were changed to positive labels in contest phase for our case. This was of course not known during the contest.

As stated in Algorithm 1, we used k-means algorithm (PROC FASTCLUS in SAS software) to do our initial classification with only one positive seed available. We randomly picked one data point as negative seed, together with the known positive seed as the initial cluster center for two classes. Only numerical features were used. Each feature was standardized by z-scaling before the clustering process. The distance between each data point and the seed is based on Euclidean distance. After the clustering process, we labeled the cluster where the positive seed lives as positive cluster, another one as negative cluster. We stored the cluster membership of each data points. Above k-means clustering process was repeated 100 times. We calculated the count of positive cluster membership of each data point after 100 iterations and used the normalized membership count as the prediction score.

During the contest phase of dataset A, we submitted 9 queries. Details of each query is listed in Table 2. We used the prediction score as uncertainty measure. The first query was carried out by randomly picking 2 data samples within the score range of 0.5 to 0.6 which is in 60 - 66 percentile of all training data samples. We got one positive label and one negative label. We used these 2 labels together with the first seed as the new seeds for the

Table 1: Label difference of dataset A (training sample) between contest phase and verification phase.

Labels in contest phase	Labels in verification phase	Frequency	Percentage
-1	-1	15579	88.85%
1	-1	688	3.92%
1	1	1267	7.23%

next round of k-means clustering. We randomly picked other unlabeled data from the pool as "negative" seeds as described in first step. We repeated our k-means process 100 times. Each time we got 2 clusters which were labeled by the known seeds. We again averaged the membership counts of each data points over 100 to create our prediction score. This semi-supervised learning process was used in the first 7 submissions. The total cost was 67 including the first seed. The query samples for submissions 2 through 6 were based on value of prediction score (in the range of 0.5 and 0.6). We started big label purchases from submission 7. Random sampling was used in these large queries. We had some concerns that the uncertainty sampling might have bias because it was based on the "current" model's prediction which was built on small number of labels. We used gradient boosting decision tree as the classifier in our submissions 8 to 10. The typical parameter setting was as following: number of trees = 700, learning rate = 0.015, subsample rate = 0.7, number of nodes = 6, minimum number of child =100, percentage of testing = 0.25.

Table 2: Queries submitted in contest phase for dataset A. The final global score = 0.45.

Submission Sequence	Number of Samples in Query	Number of Queried Samples	AUC	Sampling Strategy
1	2	1	0.54	Uncertainty and selective
2	2	3	0.61	Uncertainty and Selective
3	4	5	0.61	Uncertainty and Selective
4	9	9	0.63	Uncertainty and Selective
5	31	18	0.64	Uncertainty and Selective
6	18	49	0.63	Uncertainty and Selective
7	5030	67	0.66	Random
8	7244	5097	0.90	Random
9	5194	12341	0.92	Get all
10	0	17535	0.91	

In the verification phase, we used exactly same classification approach as the contest phase. However, we changed the query strategy. Our experience during the contest phase, taught us that there were no significant improvements in the first several small queries. In fact, it actually lowered the global score. Therefore, we decided to conduct a relatively large query in the first submission. We used the exactly same prediction score in the first submission as in contest phase. This is because the dataset was exactly same, the first seed was exactly same, only some labels were different which we did not know without querying. We obtained 233 labels after the first query which enables us to use boosting decision tree algorithm to build classification model. We used the same algorithm in the last 3 submissions. Uncertainty and selective sampling was used in the first two queries. Random sampling was used in the rest of large queries. Details of the queries are listed in Table 3. Our final global score = 0.62 which ranked 2nd in the competition. Figure 1 shows the learning curve of our submissions in verification phase.

Table 3: Queries submitted in verification phase for dataset A.

Submission Sequence	Number of Samples in Query	Number of Queried Samples	AUC	Sampling Strategy
1	232	1	0.67	Uncertainty and Selective
2	1959	233	0.82	Uncertainty and Selective
3	4286	2192	0.92	Random
4	11057	6478	0.94	Get all
5	0	17535	0.93	

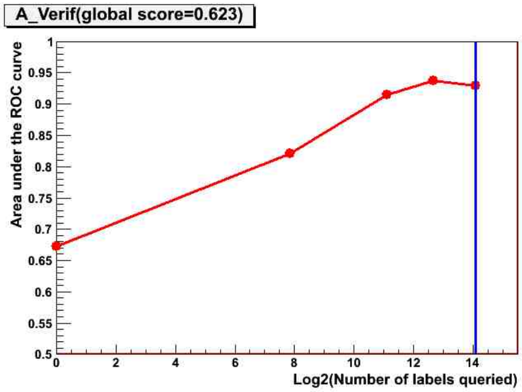

Figure 1: Learning curve of dataset A in verification phase

2.2. Dataset C: Chemo-informatics

We took the same approach on dataset C as we did for dataset A, i.e., using the stochastic semi-supervised learning process for the first submission. However, we adjusted our query strategy: we purchased all the labels in one query. This is actually a passive learning process. Our purpose is to test how good this passive learning will be comparing with the active learning carried out by others. There are 851 variables in the dataset. We did a very simple variable selection by filtering out the variables without noticeable variance (one value occupies more than 99% population) before we started our semi-superivsed clustering process. We then standardized all variables with mean of 0 and standard deviation of 1. The k-means clustering process is exactly same as that of in dataset A. The number of iterations n is set to 100. We used boosting decision tree algorithm to build the final model after the first (also the last) query. The final global score = 0.33 which ranked No.4 in the competition. Our passive learning approach did not achieve the best result.

2.3. Dataset D: Text Processing

Our approach encountered a large obstacle in dataset D. This dataset has some characteristics that are very different from other datasets.

1. It has 12000 features. This number is even bigger than the number of total training samples. Therefore, good variable selection has increased importance in order to have a better classification model.
2. It has 25.2% positive labels, while all other datasets have less than 10% positive labels.
3. The positive seed given actually has very low score which means it is on the other side of decision boundary after all labels are known.

Our method did not work well because of factors 2 and 3. However, these facts were not known during the contest. We used the same approach as in datasets A and C. One assumption in our approach is that the dataset is highly imbalanced. The positive label population is much smaller than the negative label population. This guarantees that in our stochastic process, the probability of getting negative seeds from unlabeled data pool is much higher than that of getting positive seeds. Our prediction score is based on the cluster membership counts of each data point. The cluster label is based on the given seed. The fact that the first positive seed is closer to majority of negative examples than majority of positive seeds made our first prediction score worse than random guess (the AUC of our first submission was 0.46). We did 2 small purchases of labels using uncertainty sampling guided by prediction score and then followed a large purchase to get all training labels. Our final global score is 0.33 which ranked No. 18 out of 19 participated teams in this dataset. Details can be seen in Figure 2.

2.4. Dataset B: Marketing

It is not hard to figure out during the contest that dataset B came from marketing domain. We see that the corresponding development dataset has an extremely imbalanced class distribution (positive label is only 1.78%). As stated in algorithm 1, we use logistic regression as classifier instead of k-means clustering at the beginning of the label purchase.

There were a lot of missing values in the dataset. We first did a preprocessing by simply filling the missing value with 0. We then standardized all variables (except one categorical variable, column 14) with a mean of 0 and a standard deviation of 1. There were 250 variables in the dataset. Most of them were populated by several distinct values. We did a simple unsupervised variable selection based on Shannon entropy, which is defined as

$$Entropy = -\sum_{i} p_i \log p_i \tag{1}$$

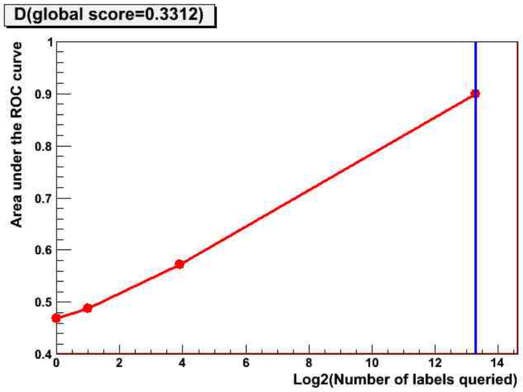

Figure 2: Learning curve of dataset D submission. Final global score = 0.33.

where p_i is the distribution of i-th bin (or i-th entry for categorical variable). We kept all the variables with entropy value greater than 0.03. This left us 43 out of 250 variables. The rationale behind this approach is to get rid of all the variables with very skewed distribution (for example, large amount of population filled by one value). All constant variables or close to constant were removed by this method.

We took the initial positive seed given by organizer, then randomly sampled 20 unlabeled data points as "negative" labels. The reason we chose 20 is because we assume the percentage of positive labels is smaller than 5% in the dataset. Actually positive labels make up 9.16% after all labels are purchased, even though the corresponding development dataset has only 1.78% positive labels. The organizer changed the class distribution purposely in order to test the robustness of every competitor's approach. We built an over-fitted logistic regression model on these 21 samples (we call it over-fit because the number of features is greater than the number of examples). This model was used to score all data points. We repeated our sampling process for "negative" labels n iterations like we did for k-means. At the end, each data point got n scores. The averaged score over n iterations ($n = 100$) was used to label the unlabeled data. We labeled all data points with score in lowest 10% as negative and in highest 10% as positive. Final logistic regression model was built on dataset with the "derived" labels. The score was used as final prediction in the first submission. We obtained two more labels after the first query. We repeated the process that we used in the first prediction. We purchased all the labels after the 2nd query. We then used gradient boosting decision tree for our final prediction. In order to make sure all good variables were included in the final model, we put back all 250 variables in the final training dataset. We did 3 rounds of bagging on boosting decision tree models. The results is shown in Figure 3. We did get the highest AUC for the final prediction score among all submissions. Our global score is 0.3754 which ranked No. 2 in the competition. The winner had a global score of 0.3757 with a passive learning approach. The winner's initial prediction had a better AUC and was their advantage.

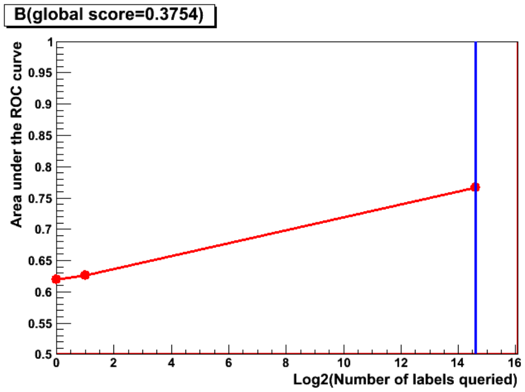

Figure 3: Learning curve of data B submission.

2.5. Dataset E: Embryology

There are 154 continuous variables in dataset E. We standardized each variable to a mean of 0 and a standard deviation of 1. We did not perform any unsupervised variable selection. We used exactly same approach as we did in dataset B. We tried a different query strategy for our first label purchase. In stead of uncertainty query we did most of the time, we chose a certainty query, i.e., we queried 2 data points in the highest 1% of the scores. The labels we obtained for those 2 data points is one positive and one negative, respectively. We repeated our stochastic semi-supervised learning process by keeping these newly purchased labels in each random sampling/learning iteration. The learning curve of dataset E is listed in Figure 4. We can see that the AUC for second submission actually becomes worse than the first submission. The newly queried labels over corrected the first model mainly due to the negative label we got which had a high prediction score of 0.75 in first model. However, it had a score of 0.03 in the second model. We did 2 more small label purchases using uncertainty sampling. The whole process is shown in Table 4. We then queried all labels and built the last model using boosting decision tree. The final global score is 0.53 which ranked No. 3 in the competition.

Table 4: Queries submitted for dataset E. The final global score = 0.53.

Submission Sequence	Number of Samples in Query	Number of Samples Queried	AUC	Sampling Strategy
1	2	1	0.75	Certainty
2	3	3	0.66	Uncertainty and Selective
3	3	6	0.67	Uncertainty and Selective
4	32243	9	0.72	Get all
5	0	32252	0.86	

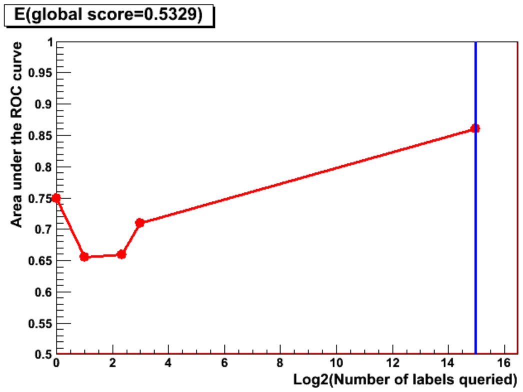

Figure 4: Learning curve of data E submission

2.6. Dataset F: Ecology

Dataset F has only 12 variables. We did not perform any variable reduction/selection. We standardized all numerical variable to a mean of 0 and a standard deviation of 1. Once again, we used same approach as in dataset E except we modified the sampling strategy. For this dataset, we skipped one small label queries and did an additional large label queries. We realized that most of the small number of label queries damaged the global score. After the 3rd submission, we obtained 552 labels. We switched from semi-supervised learning to supervised learning by using boosting decision tree algorithm. Table 5 lists each query steps. The learning curve is shown in Figure 5. Our final global score is 0.77, placed No. 4 in the competition.

Table 5: Queries submitted for dataset F. The final global score = 0.77.

Submission Sequence	Number of Samples in Query	Number of Samples Queried	AUC	Sampling Strategy
1	2	1	0.76	Uncertainty and Selective
2	7	3	0.73	Uncertainty and Selective
3	542	10	0.77	Uncertainty and Selective
4	5175	552	0.95	Random
5	61901	5727	0.98	Get all
6	0	67628	0.99	

3. Results and Discussion

Our final results and the comparison with the winners of each dataset are listed in Table 6. It is interesting to notice that six datasets have six different winners. None of the teams won more than one dataset, which means no team's method was completely general. Our

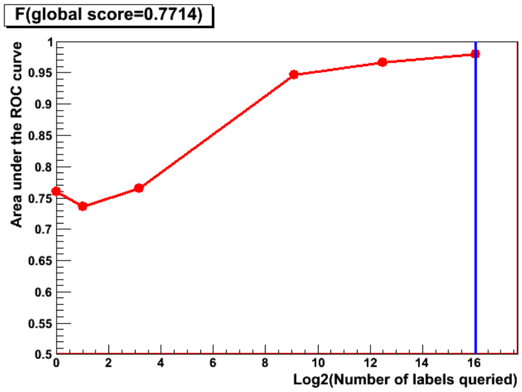

Figure 5: Learning curve of data F submission

overall ranking placed 3rd among 22 participated teams. Our approach worked relatively well on 5 out of 6 datasets. It is dataset D that degraded our overall performance. This degradation is largely because dataset D has the highest positive label percentage (25.2%, while all others are less than 10%).

Table 6: Results of each dataset and the comparison with the winners.

Data set	Positive label %	AUC	ALC	Num of queries made	Rank	Winner's AUC	Winner's ALC
A	7.23	0.9250	0.6230	4	2	0.8622	0.6289
B	9.16	0.7670	0.3754	2	2	0.7327	0.3757
C	8.15	0.8137	0.3341	1	4	0.7994	0.4273
D	25.19	0.8897	0.3312	3	18	0.9641	0.7449
E	9.03	0.8650	0.5329	4	3	0.8939	0.6266
F	7.68	0.9883	0.7714	5	4	0.9990	0.8018

We summarize some of the challenges we have seen during the competition in the following.

1. How to consistently get better performance with only a few (less than 10) known labels across different datasets. In this contest, this was very critical because of the way how the global score was calculated and the log 2 scaling on number of queried samples. All top players of each dataset had good performance at the first submission. It is very hard to find a robust method working for all dataset. That is the primary reason why nobody won more than one dataset. Our stochastic approach attempts to improve the robustness of our method. It works relatively well overall.

2. How to consistently improve model performance with the increase of known labels in a given dataset. This is particularly hard when the number of known labels is small.

We saw cases both in our own submission and in others that the model performance was getting worse when a few more labels were added into the existing model. This was because the initial model was heavily impacted by the initial labels. A few newly added labels may not be representative to the whole dataset, especially when the uncertainty sampling query is used.

3. It is not conclusive that active learning approach will always beat passive learning in these real world data sets based on the current global score measurement, particularly when the data dimension is high and the label distribution is imbalanced. Winners of dataset B, D and E all used passive learning. The $\log 2$ scaling in the global score calculation might give too much weight to models with only a few labels. The passive learning approach avoids the dips of learning curve and gains some "artificial" advantages when the dataset is hard to learn.

4. Conclusion

In summary, we have described the method we used in all six datasets in the active learning challenge. We propose a stochastic semi-supervised learning approach to tackle the classification problem under the condition that the number of labeled data points is extremely small and the two classes are highly imbalanced. We prefer to use k-means clustering for datasets with a very large number of features (greater than 800 in this contest). We suggest using logistic regression for datasets with highly imbalanced class distribution. In the contest, we switched to supervised learning using gradient boosting decision tree algorithm when the number of known labels is greater than 200, which corresponds to the middle range of x value in the area under the learning curve. Both uncertainty sampling and density-based selective sampling were used for label queries. Our approach performed pretty well in 5 out of 6 datasets. We ranked 3rd overall in the contest.

Acknowledgments

We thank the organizers of the active learning challenge for setting up an excellent platform and providing real world datasets for this competition. We thank David Zaleta for proof reading the manuscript. J. Xie would like to thank Dr. Isabelle Guyon for providing him the travel support.

References

Y. Baram, R. El-Yaniv, and K. Luz. Online choice of active learning algorithms. *JMLR*, pages 255–291, 2004.

A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. *Proceedings of the 11th Annual Conference on Computational Learning Theory*, pages 92–100, 1998.

J. F. Bowring, J. M. Rehg, and M. J. Harrold. Active learning for automatic classification of software behavior. *In Proceedings of the International Symposium on Software Testing and Analysis*, pages 195–205, 2004.

R. Dara, S. Kremer, and D. Stacey. Clustering unlabeled data with soms improves clssification of labeled real-world data. In *Proceedings of the World Congress on Computation Intelligence(WCCI)*, May 2002.

A. Demiriz, K. Bennett, and M. Embrechts. Semi-supervised clustering using genetic algorithms. In *Proceedings of Artificial Neural Networks in Engineering*, November 1999.

Y. Freund, H. S. Seung, E. Shamir, and N. Tishby. Selective sampling using the query by committee algorithm. *Machine Learning*, 28:133–168, 1997.

Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29:1189–1232, 1999.

Isabelle Guyon, Gavin Cawley, Gideon Dror, and Vincent Lemaire. Results of the active learning challenge. *JMLR W& CP*, this volume.

D. D. Lewis and W. A. Gale. A sequential algorithm for training text classifiers. *SIGIR*, pages 3–12, 1994.

T. Luo, K. Kramer, and D. B. Goldgof. Active learning to recognize multiple types of plankton. *JMLR*, 6:589–613, April 2005.

Charles Rosenberg, Martial Hebert, and Henry Schneiderman. Semi-supervised self-training of object detection models. In *Seventh IEEE Workshop on Applications of Computer Vision*, January 2005.

N. Roy and A. McCallum. Toward optimal active learning through sampling estimation of error reduction. *ICML*, pages 441–448, 2001.

Burr Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

S. Tong and D. Koller. Support vector machine active learning with applications to text classification. *JMLR*, 2:45–66, November 2001.

Xiaojin Zhu. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2005.