Semi-supervised Learning from only Positive and Unlabeled Data using Entropy

Xiaoling Wang[1], Zhen Xu[2], Chaofeng Sha[2], Martin Ester[3], and Aoying Zhou[1,2]

[1] Software Engineering Institute, East China Normal University, China
[2] Shanghai Key Lab. of Intelligent Information Processing, Fudan University, China
[3] School of Computing Science, Simon Fraser University, Burnaby, Canada
{xlwang,ayzhou}@sei.ecnu.edu.cn
{xuzhen,ayzhou}@fudan.edu.cn

Abstract. The problem of *classification from positive and unlabeled examples* attracts much attention currently. However, when the number of unlabeled negative examples is very small, the effectiveness of former work has been decreased. This paper propose an effective approach to address this problem, and we firstly use entropy to selects the likely positive and negative examples to build a complete training set; and then logistic regression classifier is applied on this new training set for classification. A series of experiments are conducted. The experimental results illustrate that the proposed approach outperforms previous work in the literature.

1 Introduction

The traditional classification task is to construct a classification function $y = \varphi(x)$, which maps any instance $x \in X$(instance space) to one known class label $y \in \{c_1, \cdots, c_n\}$(class label set), based on the labeled training set. This kind of classification method is the *supervised learning*, and it often requires a great deal of labeled instances for training; however, in many real applications, it is impossible to provide enough training examples. As a result, *semi-supervised learning*, i.e. *learning from labeled and unlabeled examples*, is studied, where only a few labeled examples together with large number of available unlabeled ones are given. Different from *learning from positive and negative examples*, another special kind of the problem, namely, *learning from* ***positive*** *and* ***unlabeled*** *examples*, gains more and more attention. It deals with such a case that no labeled negative examples exist in the training set, that is to say, only a few labeled positive examples and lots of unlabeled examples are available, without any information about negative class.

Take email classification for an examples. Assuming that there exist 2 different classes of mails: news and entertainment. We can build a classifier according to the content of mails. By this classifier, the incoming mail about the news and entertainment can be classified easily and correctly. However, if the new incoming mail is about job description, it is not belonging to any defined classes. As a result, assigning it to any one of the 2 classes, it is obviously unsuitable. Instead,

it should be assigned to the negative class, different from the given two positive classes.

This paper addresses the problem of *learning from positive and unlabeled examples*, where the input are multiple(at least 2) labeled positive training classes and unlabeled data. We design an entropy based method to obtain some information of unlabeled examples. For this kind of problem, several related approaches are proposed in recent years. There are mainly two different kinds as follows:

- Completely discarding the unlabeled examples. E.g., one-class SVM[1] constructs a size-suitable area, approximately covering all labeled positive examples, beyond which are negative examples. However, the trained classifier only learns from the labeled positive examples but knows nothing about the negative examples, which easily makes over-fitting.
- Another kind of approaches, e.g., PEBL[2] and Roc-Clu-SVM[3], consider how to find the most likely negative examples, add them to the training dataset, and then apply a classical classification method; LGN[4] is to generate negative examples.

Different from former work, this paper proposes a entropy-based method - **SLE**(**S**emi-supervised **L**earning from only positive and unlabeled data using **E**ntropy). The contribution of this paper is as follows:

1. We study the problem of classifying unlabeled data using positive training set and present an semi-supervised learning method to solve this problem.
2. In order to discover the hidden information in the unlabeled examples, **Entropy** is adopted to find out the most likely positive or negative examples to build a new training set which contains information of unlabeled examples.
3. After constructing the new training set by **Entropy**, a logistic regression classifier is directly used for the classification task. Other traditional classifiers, such as SVM, can also be used for the rest classification task based on new training set.
4. A series of experiments are conducted. We test the effectiveness on two different benchmarks, using F-score as measure. Experimental results verify that our method outperforms previous work.

This paper is organized as follows: Section 2 presents existing related research works; Section 3 introduces the proposed **SLE** approach; Finally, The experimental results are in Section 4, followed by the conclusion in Section 5.

2 Related Work

There are four kinds of approaches for this problem.

1. One simple approach is to completely ignore the unlabeled examples, namely, learning only from the labeled positive examples. E.g., one-class SVM[1] is to construct a suitable area, approximately covering all labeled positive examples. Beyond this region, all examples are viewed as negative. Obviously, it

probably discards very useful information hidden in the unlabeled set, which may make much more contribution to classify or discriminate. Therefore, it easily makes over-fitting.

2. With the unlabeled examples, co-training based on labeled positive ones is one other kind of popular current approach. It usually uses two-phases strategy: (i) extracting likely negative examples from the unlabeled set; (ii) simply using traditional learning methods for the rest classification task. For example, PEBL[2] firstly utilizes 1-DNF[5] to find out quite likely negative examples and secondly employs SVM[6] iteratively for classification. Roc-Clu-SVM[3] uses clustering technique for extraction of likely negative examples in the first step, and sequently SVM[6] is applied for the rest classification task.
3. Quite different from the above ones, B-Pr[7] and W-SVM[8] belong to the probabilistic approaches. Without needing to extract likely negative examples, they transform the original problem $p(y|x)_{y\in\{+,-\}}$ into a simple sample model $p(t|x)_{t\in\{P,U\}}$, where P is the training set and U is the unlabeled data set. But there are several parameters having to be estimated, which directly affect the final performance. Additionally, they can't cope with the extreme case of imbalance existing in the unlabeled set.
4. LGN[4] is based on the assumption that the identical distribution of words belonging to the positive class both in labeled positive examples and unlabeled ones, it generates an artificial negative document for learning. However, in practice, it is very hard or impossible to satisfy this assumption.

Different from the above approaches, we use the entropy's property to reveal the hidden information in the positive instances, and then to find the likely positive and negative instances.

3 The Proposed Approach: SLE

This section introduces the proposed **SLE**(**S**emi-supervised **L**earning method using **E**ntropy) classification method to solve the problem where the training data only contain positive and unlabeled examples. Preliminaries are given in Section 3.1.

3.1 Preliminaries

Feature Extraction. Each document object is modeled as a feature vector with hundreds or thousands of term dimensions. TF-IDF are borrowed from IR to measure the weigh of each term.

$$\text{TF-IDF}_{(t,c_i)} = \frac{\sum_{d\in c_i} n_t}{\sum_{d\in c_i}\sum_{w\in d} n_w} \times \lg\frac{|D|}{|D_t|}$$

where t and w are word features; c_i is the i-th class in training set; n_t and n_w are feature frequency, i.e., times of word t and w occurring in object d respectively;

$|D|$ is the number of all objects; $|D_t|$ is the number of objects containing word t.

Over-Sampling. Re-sampling is a commonly used technique in traditional classification method in order to handle the imbalanced examples in different classes. There are two methods: under-sampling and over-sampling. Generally, through under-sampling, the size of major classes decreases to the scale of minor ones; Oppositely, over-sampling makes the number of minor classes' examples approximate to the number of major ones.

Based on the assumption that more examples implies more information, more contribution to classify. Over-sampling are adopted, and original set S and over-sampled set S^o must meet the following relationship: $S^o \supseteq S$.

Entropy $H(x)$**.** The entropy [9] of the probability distribution $P(x)$ on the set $\{x_1, \cdots, x_n\}$ is defined as:

$$H(x) = -\sum_{i=1}^{n} p(x_i) \lg p(x_i)$$

In this paper, we calculate the entropy of posterior probabilities of every instance d_i in the unlabeled set, and then the revised formula of entropy is defined as follows:

$$H(d_i) = -\sum_{j=1}^{|C|} p(c_j|d_i) \lg p(c_j|d_i)$$

where $p(c_j|d_i)$ is the class posterior probability of instance d_i belonging to the j-th class c_j; $|C|$ is the number of known or predefined class labels appearing in the training set.

For the unlabeled set(labeled as U), the entropy of every instance are calculated, and the instances

$$d_i = \arg\min_{d_i \in \{d_t | \arg\max_{c_k \in C}(p(c_k|d_t))\}} (H(d_i))$$

and

$$d_j = \arg\max_{d_j \in U} (H(d_j))$$

are obtained as the most likely corresponding sub-positive(c_k) and negative examples respectively.

3.2 Problem Description

Given that a training set(P) only containing positive examples, from multiple (at least 2) classes, without negative ones, and one unlabeled set(U) containing both positive and negative examples, our task is to construct a classifier(C), finding all likely negative examples(U_n) hidden in the unlabeled set and it is formulated as follows: input$(P, U) \overset{C}{\Rightarrow}$ output(U_n).

Algorithm 1: Find-Positive

Input: training set P, unlabeled set U
Output: positive instances set S_p

1. C=$\{c_1, \cdots, c_n\}$ are all positive class labels in P;
2. $S_p = \emptyset$;
3. **for** each set B_k,$1 \leq k \leq n$ **do**
4. $B_k = \emptyset$;
5. **endfor**
6. **for** each instance $d_i \in U$ **do**
7. $H(d_i) = -\sum_{j=1}^{|C|} p(c_j|d_i) \lg p(c_j|d_i)$;
8. $k = \arg\max_{c_k \in C}(p(c_k|d_i))$;
9. $B_k = B_k \cup \{d_i\}$;
10. **endfor**
11. $topk = \lambda \times \min_{c_k \in C}|B_k|$;
12. **for** each set B_k,$1 \leq k \leq n$ **do**
13. **while** $B_k.size() > topk$ **do**
14. $d_i = \arg\max_{d_i \in B_k}(H(d_i))$;
15. $B_k.remove(d_i)$;
16. **endwhile**
17. **endfor**
18. **for** each set B_k,$1 \leq k \leq n$ **do**
19. S_p=$S_p \cup B_k$;
20. **endfor**
21. **return** S_p;

Fig. 1: Finding likely positive examples

3.3 SLE Approach

The detailed algorithm are shown in Algorithm 1, 2 and 3.

- In Algorithm **1**, the original training set(P) is used to learn a classifier(FP) for classifying the unlabeled set(U), and then find all likely positive classes examples(S_p) for further learning, and formulate it as: $input(P,U) \overset{\mathbf{FP}}{\Rightarrow} output(S_p)$, where P and U is the input, S_p is the output.
- In Algorithm **2**, the new training set($P + S_p$) are used to learn a new classifier(FN) for classifying the new unlabeled set($U - S_p$), and then find likely negative examples(S_n) as training data: $input(P + S_p, U - S_p) \overset{\mathbf{FN}}{\Rightarrow} output(S_n)$, where $P + S_p$ and $U - S_p$ is the input, S_n is the output.
- In Algorithm **3**, $P + S_p$ is the positive class and S_n is the negative class for learning, i.e. there exist only two classes(positive and negative) in the training set. Then this new training set is used to learn a logistic classifier(SLE), and to classify the new unlabeled set to find out all negative examples(U_n):

$input(\{P+S_p, S_n\}, \{U-S_p-S_n\}) \overset{\textbf{SLE}}{\Rightarrow} output(U_n)$, where $\{P+S_p, S_n\}$ and $\{U-S_p-S_n\}$ is the input, U_n is the output.

Algorithm 2: Find-Negative

Input: training set P, unlabeled set U
Output: negative instances set S_n

1. C=$\{c_1, \cdots, c_n\}$ are all positive class labels in P;
2. $S_n = \emptyset$;
3. **for** each set B_k,$1 \leq k \leq n$ **do**
4. $B_k = \emptyset$;
5. **endfor**
6. **for** each instance $d_i \in U$ **do**
7. $H(d_i) = -\sum_{j=1}^{|C|} p(c_j|d_i) \lg p(c_j|d_i)$;
8. **if** $S_n.size() < \mu$ **then**
9. $S_n = S_n \cup \{d_i\}$;
10. **else if** $H(d_i) > \min_{d_k \in S_n} (H(d_k))$ **then**
11. $S_n.remove(d_k)$;
12. $S_n = S_n \cup \{d_i\}$;
13. **endif**
14. **endif**
15. **endfor**
16. **return** S_n;

Fig. 2: Finding likely negative examples

In the above algorithms, we adopt the logistic regression classifier [8, 10]. Obviously, other traditional classifiers can also been used in the presented method. In the experiments, we use three standard classifiers published in weka[4], including SVM[11], RLG[12] and LMT[13, 14], as the classifiers in the ***Algorithm 3*** of SLE approach.

4 Experimental Evaluation

In this section, we evaluate the performance of our approach. The experiments are conducted on Intel 2.6 GHZ PC with 2 GB of RAM.

The objective of experiments are listed here:

– Firstly, we study the performance of SLE over different data sets including text and nominal data.

[4] http://www.cs.waikato.ac.nz/ml/weka

Algorithm 3: SLE

Input: training set P, unlabeled set U
Output: negative instances set U_n

1. C=$\{c_1, \cdots, c_n\}$ are all positive class labels in P;
2. $U_n = \emptyset$;
3. P = P $\cup$ Find-Positive(P, U);
4. U = U - Find-Positive(P, U);
5. P = P $\cup$ Find-Negative(P, U);
6. U = U - Find-Negative(P, U);
7. Merge all positive classes into one big positive class, i.e, training set has only two classes: one positive class("+") and one negative class("-");
8. **if** P is unbalanced **then**
9. over-sample P for making it balanced;
10. **endif**
11. **for** each instance $d_i \in U$ **do**
12. **if** $p(-|d_i) > p(+|d_i)$ **then**
13. $U_n = U_n \cup \{d_i\}$;
14. **endif**
15. **endfor**
16. **return** U_n;

Fig. 3: SLE classification algorithm

– Secondly, we test the robustness of the proposed method for the different number of negative examples, i.e. when the ratio of negative examples varies in the unlabeled data set.
– Thirdly, we verify the importance of *Entropy* to deal with the extremely imbalanced unlabeled set by evaluating Algorithm **1**. If the number of unlabeled negative examples is very small in the total unlabeled data set, using Algorithm 1 to find the likely positive examples firstly are very helpful.

Two representative real benchmarks are used. The two benchmark information is listed in Table 1.

– The first benchmark is 20 Newsgroups[5], where the data covers many different domains, such as **computer**, **science** and **politics**.
– The second benchmark is UCI repository[6]. In this paper, we use the **letter** data set, which identifies each black-and-white rectangular pixel display one of the 26 capital letters in the English alphabet. This benchmark is to test the SLE method for nominal data, not text data type.

We implement the proposed SLE method in Section 3 and three former approaches: LGN[4], NB-E[15] and one class-SVM[1]. As discussed in Section 3.3, SLE consists of **L-SVM**, **L-RLG** and **L-LMT** respectively.

[5] http://people.csail.mit.edu/jrennie/20Newsgroups
[6] http://archive.ics.uci.edu/ml/datasets.html

Table 1: The characters of two benchmark

	dataSource	**#inst.**	**#attr.**	**#class**	**type**
Benchmark1	20Newsgroups	20,000	>100	20	text
Benchmark2	UCI Letter	20,000	16	26	not-text

1. L-SVM adopts the standard SVM[11] as a classifier in the Algorithm 3 of SLE method.
2. L-RLG uses a logistic regression classifier RLG[12] in the Algorithm 3, which uses ridge estimators[16] to improve the parameter estimates and to diminish the error of future prediction.
3. L-LMT applies LMT[13, 14] as the basic classifier in the Algorithm 3 of SLE method, which combines tree induction methods and logistic regression models for classification.

4.1 Experimental Setting

Classification Task.

For two data collections, we define the classification tasks as follows.

- ***Benchmark 1***. 20 Newsgroups has approximately 20000 documents, divided into 20 different small subgroups respectively, each of which corresponds to a different topic. We firstly choose four subgroups from **computer** topic and two topics from **science** respectively, i.e. {comp.graphics, comp.ibm.hardware, comp.mac.hardware, comp.windows.x}×{sci.crypt, sci.space}, totally $C_4^1 C_2^1 = 8$ pairs of different experiments.
  2-classes problem: For each pair of classes, i.e. selecting one class from {graphics, ibm.hardware, mac.hardware, windows.x}×{crypt, space} respectively as two positive classes, e.g. **graphics**×**crypt**, a equal part of documents are chosen randomly for training as corresponding positive instances, and the rest as unlabeled positive data in unlabeled set; Then some examples are extracted randomly from the rest 18 subgroups are viewed as unlabeled negative examples in unlabeled set, and the number is $\alpha \times |U|$, where α is a proportion parameter, showing the percentage of negative examples in the unlabeled set, and $|U|$ is the number of all instances in the unlabeled set.
  3-classes problem: Similar to the above 2-classes experiment, except that the third positive class is randomly chosen from the rest 18 classes.
- ***Benchmark 2***. UCI Letter data is used in the experiment, which contains approximately 20000 instances, divided into 26 different groups, i.e. from **a** to **z**. For simplicity, we divide them into two parts $\{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}\}\times\{\mathbf{O}, \mathbf{P}, \mathbf{Q}\}$, totally $C_4^1 C_3^1 = 12$ pairs.
  2-classes problem: Any pair firstly select one class from $\{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}\}\times\{\mathbf{O}, \mathbf{P}, \mathbf{Q}\}$ respectively as two positive classes, e.g. $\mathbf{A}\times\mathbf{O}$, and the rest settings are the same to the above experiments.
  3-classes problem: It is similar to the above experiments.

4.2 Experimental Result

The experiment randomly runs six times to get the average F-score value as the final result. In the experiments, α is the ratio of the unlabeled negative examples compared to the unlabeled set. E.g. $\alpha = 0.05$ means that the number of the unlabeled negative examples is only 5% of the unlabeled set.

Performance for the 2-classes problem.
Table **2** records the F-score values computed by 1-SVM, LGN, NB-E, L-SVM, L-RLG and L-LMT on the benchmark1, i.e., 20Newsgroups dataset. As shown in Table 2, for each row, L-SVM, L-RLG and L-LMT are better than other classifier methods; Meanwhile, NB-E outperforms LGN, but both NB-E and LGN are better than 1-SVM.

Table 2: Performance of 2-classes problem over Benchmark1 ($\alpha = 0.05$)

20newsgroup data	**1-SVM**	**LGN**	**NB-E**	**L-SVM**	**L-RLG**	**L-LMT**
graphics - crypt	0.041	0.321	0.55	0.768	0.591	0.73
graphics - space	0.033	0.324	0.464	0.807	0.592	0.835
ibm.hardware - crypt	0.034	0.4	0.421	0.651	0.674	0.715
ibm.hardware - space	0.04	0.366	0.562	0.835	0.723	0.882
mac.hardware - crypt	0.041	0.444	0.557	0.816	0.866	0.856
mac.hardware - space	0.048	0.374	0.604	0.784	0.657	0.83
windows.x - crypt	0.025	0.365	0.422	0.703	0.7	0.788
windows.x - space	0.025	0.264	0.592	0.735	0.531	0.694
average	**0.03**	**0.35**	**0.52**	**0.76**	**0.66**	**0.79**

From the experimental results, we can see that: (1)When the precondition that the identical distribution of positive words in the training set and unlabeled set is not met, LGN do not perform well; (2)When the number and purity of training examples is not big or high, 1-SVM has unsatisfactory performances, even worse; (3) For both text data and nominal data, L-SVM, L-RLG and L-LMT have much better performance than others approaches.

In Fig.**4(C)**, there are not any positive and negative words existing in UCI letter, completely violating the assumption of LGN, therefor its F-score values of LGN approach are nearly zero. Compared to document data, UCI letter has only 16 attributes, hence the performance of all classifier are relatively trivially poor, especially 1-SVM, L-SVM and LGN.

Performance for the 3-classes problem.
Table 3 records the F-score values of different classification approaches over 3-classes UCI letter data. As shown in Table 3, L-RLG and L-LMT are better than other classifier methods; Meanwhile, F-score values of LGN are zero, but 1-SVM better than NB-E.

The experimental results of 3-classes problem are shown in Fig.**4(B)** and **(D)**. In the 3-classes experiment, every classifier still remains consistent performance with in the 2-classes experiments.

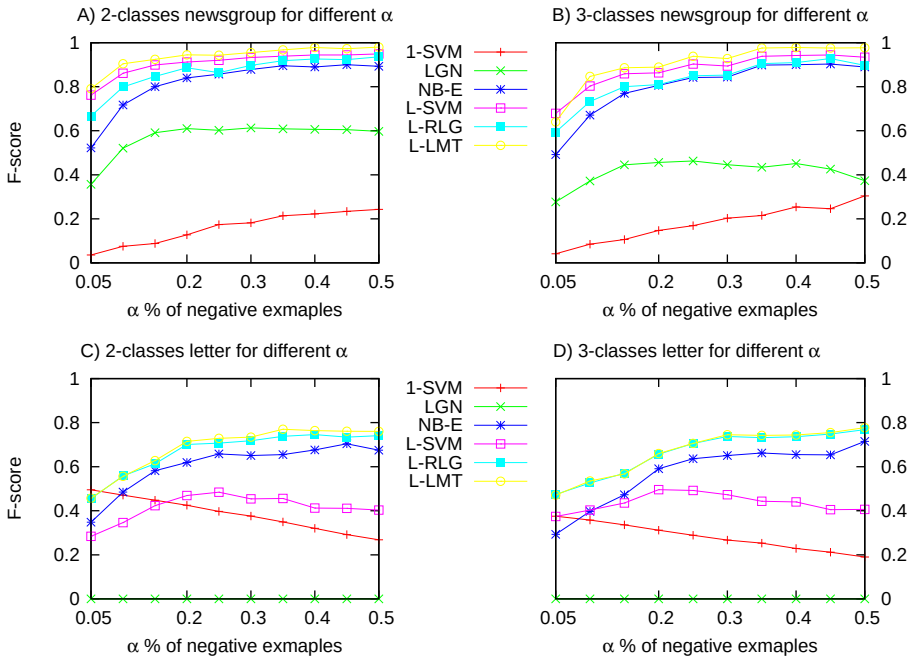

Fig. 4: F-score values for different α

Table 3: Performance of 3-classes problem over Benchmark2($\alpha = 0.05$)

uci-letter data	**1-SVM**	**LGN**	**NB-E**	**L-SVM**	**L-RLG**	**L-LMT**
A - O	0.316	0	0.239	0.346	0.392	0.395
A - P	0.449	0	0.343	0.336	0.528	0.548
A - Q	0.38	0	0.248	0.427	0.547	0.54
B - O	0.294	0	0.386	0.413	0.568	0.56
B - P	0.462	0	0.426	0.493	0.549	0.537
B - Q	0.376	0	0.26	0.383	0.52	0.491
C - O	0.305	0	0.328	0.305	0.505	0.481
C - P	0.427	0	0.241	0.417	0.453	0.43
C - Q	0.37	0	0.266	0.358	0.353	0.353
D - O	0.289	0	0.27	0.367	0.455	0.445
D - P	0.44	0	0.259	0.328	0.395	0.419
D - Q	0.387	0	0.257	0.313	0.42	0.453
average	**0.37**	**0**	**0.29**	**0.37**	**0.47**	**0.47**

Effect for dealing with unbalanced data Algorithm 1 finds the likely positive examples firstly. This is very helpful, when the number of unlabeled negative examples is very small in the unlabeled data set, i.e., α is small. The reason is

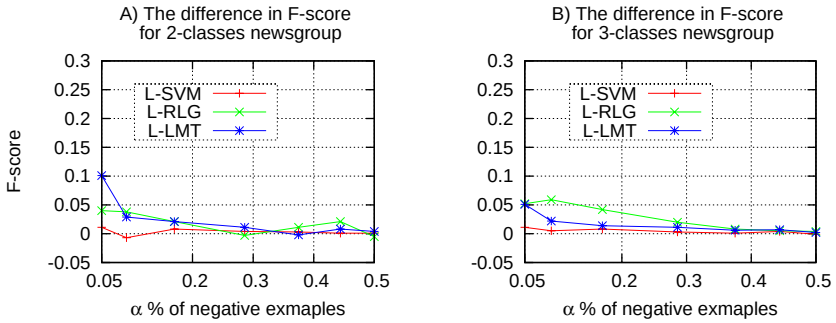

Fig. 5: Effect of Algorithm 1 with unbalanced data

Algorithm 1 improves the probability of find the negative examples in the next step. We verify this point in this section.

Fig.**5(A)** records the results of using Algorithm 1 or not using it. The goal of Algorithm 1 is to find likely positive examples. When $\alpha \leq 0.3$, i.e., the number of the unlabeled negative examples is 30% of the unlabeled set, the difference value is bigger than zero, namely, Algorithm 1 improves the final performance, especially for L-LMT and L-RLG. This results verify that Algorithm 1 is helpful to deal with unbalanced unlabeled set. Fig.**5(B)** shows the similar results in 3-classes document experiments.

From the above experiments, we can draw the following conclusion.

(1)The distribution of positive terms in the training set is often not identical to the distribution in the unlabeled set, obviously it does not meet the assumption of LGN, which is the main reason of lower F-score of LGN classification approach;

(2)NB-E approach classifies the unlabeled set base on entropy maximization, which requires a large number of unlabeled negative examples, therefor, when α grows up, the performance approximates L-RLG.

(3)Compared to other existing methods, the gain of SLE is much larger for 20 newsgroups than for letter data. There exists one main cause: for letter data, the number of attributes is only 16, but 20 newsgroups data has hundreds of attribute words, which verify that much more information will give more contribution to the classification results.

In summary, the proposed classification approach SLE outperforms other approaches, including LGN, NB-E and 1-SVM. By adopting Entropy, oversampling and logistic regression, when the number of positive classes in training set, it almost has no influence on the final classification performance.

5 Conclusion

In this paper, we tackle the problem of learning from positive and unlabeled examples and present a novel approach called **SLE**. Different from former work,

it firstly finds out likely positive examples and identifies the negative examples hidden in unlabeled set, followed by any traditional classifier for the rest task. By a series of experiments, we verify that the proposed approach outperforms former work in the literature. In the further work, we will further study the parameter learning problem and some optimization strategies to improve SLE approach.

Acknowledgments. This work is supported by NSFC grants (No. 60773075 and No. 60925008), National Hi-Tech 863 program under grant 2009AA01Z149, 973 program (No. 2010CB328106), Shanghai International Cooperation Fund Project (Project No.09530708400) and Shanghai Leading Academic Discipline Project (No. B412).

References

1. Larry M. Manevitz, Malik Yousef, Nello Cristianini, John Shawe-taylor, and Bob Williamson. One-class svms for document classification. *Journal of Machine Learning Research*, 2:139–154, 2001.
2. Hwanjo Yu, Jiawei Han, and KCC Chang. Pebl: Positive example based learning for web page classification using svm. In *KDD*, 2002.
3. Xiaoli Li and Bing Liu. Learning to classify texts using positive and unlabeled data. In *IJCAI*, 2003.
4. Xiaoli Li, Bing Liu, and See Kiong Ng. Learning to identify unexpected instances in the test set. In *IJCAI*, 2007.
5. F. Denis. Pac learning from positive statistical queries. In *ALT*, 1998.
6. C. Cortes and V. Vapnik. Support vector networks. *Machine Learning*, 20:273–297, 1995.
7. Dell Zhang and Wee Sun Lee. A simple probabilistic approach to learning from positive and unlabeled examples. In *UKCI*, 2005.
8. Charles Elkan and Keith Noto. Learing classifiers from only positive and unlabeled data. In *KDD*, 2008.
9. T. Cover and J. Thomas. *Elements of Information Theory*. Wiley Interscience, 1991.
10. Y. Guo and R. Greiner. Optimistic active learning using mutual information. In *IJCAI*, 2007.
11. C. C. Chang and C. J. Lin. Libsvm: a library for support vector machines. *http://www.csie.ntu.edu.tw/ cjlin/libsvm*, 2001.
12. Le Cessie S. and Van Houwelingen J.C. Ridge estimators in logistic regression. *Applied Statistics*, 41:191–201, 1997.
13. M. Sumner, E. Frank, and M. Hall. Speeding up logistic model tree induction. In *the 9th European Conference on Principles and Practice of Knowledge Discovery in Databases*, 2005.
14. N. Landwehr, M. Hall, and E. Frank. Logistic model trees. In *ECML*, 2003.
15. Chaofeng Sha, Zhen Xu, Xiaoling Wang, and Aoying Zhou. Directly identify unexpected instances in the test set by entropy maximization. In *APWEB-WAIM*, 2009.
16. D.E. Duffy and T.J. Samtmer. On the small sample properties of norm-restricted maximum likelihood estimators for logistic regression models. In *Communs Statist. Theory Meth.*, 1989.