A Novel Reliable Negative Method Based on Clustering for Learning from Positive and Unlabeled Examples

Bangzuo Zhang[1,2] and Wanli Zuo[1]

[1] College of Computer Science and Technology, Jilin University, ChangChun, 130012, China
[2] College of Computer, Northeast Normal University, ChangChun, 130024, China
zhangbz@nenu.edu.cn, wanli@mail.jlu.edu.cn

Abstract. This paper investigates a new approach for training text classifiers when only a small set of positive examples is available together with a large set of unlabeled examples. The key feature of this problem is that there are no negative examples for learning. Recently, a few techniques have been reported are based on building a classifier in two steps. In this paper, we introduce a novel method for the first step, which cluster the unlabeled and positive examples to identify the reliable negative document, and then run SVM iteratively. We perform a comprehensive evaluation with other two methods, and show experimentally that it is efficient and effective.

Keywords: Semi-Supervised Learning, Text Classification, Bisecting k-means Clustering, Learning from Positive and Unlabeled Examples (LPU).

1 Introduction

With the ever-increasing volume of text data from various online sources, it is an important task to categorize or classify these text documents into categories that are manageable and easy to understand. Text categorization or classification aims to automatically assign text documents to pre-defined classes. In supervised learning, text classifier relies on labeled training examples. For binary problems, positive and negative examples are mandatory for machine learning. The main bottleneck of building such a classifier is that a large, often prohibitive, number of labeled training documents are needed. But, for many learning task, labeled examples are rare while numerous unlabeled examples are easily available.

Recently, semi-supervised learning algorithms from a small set of labeled data with the help of unlabeled data have been defined. These techniques alleviate some labor-intensive effort. Semi-supervised learning includes two main paradigms: (1) learning from a small set of labeled examples and a large set of unlabeled examples; and (2) learning from positive examples and unlabeled examples (with no labeled negative examples). Many researchers have studied learning in the first paradigm [1]. In learning from positive and unlabeled examples, some theoretical studies and practical algorithms have been reported in [2-9].

In this paper, we study learning from positive data with the help of unlabeled data, which is also common in practice. For instance, in many text mining tasks, such as document retrieval and classification, one goal is the efficient classification and re-

H. Li et al. (Eds.): AIRS 2008, LNCS 4993, pp. 385–392, 2008.
© Springer-Verlag Berlin Heidelberg 2008

trieval of interests of some users. Positive information is readily available and unlabeled data can easily be collected. One example is learning to classify web page as "interesting" for a specific user. Documents pointed by the user's bookmarks defined a set of positive examples because they correspond to interesting web pages for him and negative examples are not available at all. Nonetheless, unlabeled examples are easily available on the World Wide Web.

Theoretical results show that in order to learn from positive and unlabeled data, it is sometimes sufficient to consider unlabeled data as negative ones [2-3]. Recently, a few algorithms were proposed to solve the problem. One class of algorithms is based on a two-step strategy as follow. These algorithms include Roc-SVM [7], S-EM [8], PEBL (Positive Examples Based Learning) [9].

Step 1: Identifying a set of reliable negative documents from the unlabeled set. In this step, S-EM uses a Spy technique, PEBL uses a technique called 1-DNF, and Roc-SVM uses the Rocchio algorithm.

Step 2: Building a set of classifiers by iteratively applying a classification algorithm and then selecting a good classifier from the set. In this step, S-EM uses the Expectation Maximization (EM) algorithm with a NB (Naive Bayesian) classifier, while PEBL and Roc-SVM use SVM (Support Vector Machine). Both S-EM and Roc-SVM have some methods for selecting the final classifier. PEBL simply uses the last classifier at convergence.

The underlying idea of these two-step strategies is to iteratively increase the number of unlabeled examples that are classified as negative while maintaining the positive examples correctly classified. This idea has been justified to be effective for this problem [8].

In this paper, we first introduce another method for the first step, i.e. cluster the positive and unlabeled examples to identify the reliable negative document, and evaluate our method with other two methods (PEBL, and Roc-SVM).

The remainder of this paper is organized as follow: We would like to first review the existing reliable negative methods to this problem in section 2; propose a novel clustering based approach in section 3; and comparative experiments have been made in section 4; finally make conclusion in section 5.

2 Related Works

In this section, we introduce algorithms for the first step that based on the two-step strategy. The techniques of the Roc-SVM, the S-EM and the PEBL have been reported in [7], [8], [9] respectively.

In this paper, we use P to denote the positive examples set, U for unlabeled examples set, and RN for reliable negative examples set that produced from the unlabeled examples set U.

Li, X.L. et al. report the Spy technique in the S-EM [7]. It first randomly selects a set S of positive documents from P and put them in U. Documents in S act as "spy" documents. The spies behave similarly to the unknown positive documents in U. Hence they allow the algorithm to infer the behavior of the unknown positive documents in U. In step 2, it then run EM to build the final classifier. Since NB is not a strong classifier for text classification, so we do not compare with it. This algorithm performs stably when the positive examples set is very small. When the positive examples set is large, it is worse than others.

The Roc-SVM algorithm uses the Rocchio method to identify a set *RN* of reliable negative documents from *U*. Rocchio is an early text classification method. In this method, each document is represented as a vector, Let *D* be the whole set of training documents, and C_j be the set of training documents in class *j*. Building a Rocchio classifier is achieved by constructing a prototype vector $\vec{C}_j$ for each class *j*. In classification, for each test document *td*, it uses the cosine similarity measure to compute the similarity of *td* with each prototype vector. The class whose prototype vector is more similar to *td* is assigned to *td*.

$$\vec{C}j = \alpha \frac{1}{|Cj|} \sum_{\vec{d} \in Cj} \frac{\vec{d}}{\|\vec{d}\|} - \beta \frac{1}{|D - Cj|} \sum_{\vec{d} \in D - Cj} \frac{\vec{d}}{\|\vec{d}\|}. \quad (1)$$

When use this method, the amount of *RN* is so big that biased the classifier of step 2 and poor performance, especially when the *P* set is small.

The PEBL uses the 1-DNF method, first builds a positive feature set *PF* which contains words that occur in the positive examples set *P* more frequently than in the unlabeled examples set *U*. Then it tries to filter out possible positive documents from *U*. A document in *U* that does not have any positive feature in *PF* is regarded as a strong negative document. In this algorithm, the amount of *RN* set is always small and sometimes is short text examples. Its performance is poor when the number of positive examples set is small. When the positive examples set is large, it becomes more stable.

3 The Proposed Technique

In this section, we introduce a new method for the first step that use clustering to identify a set *RN* of reliable negative documents from the unlabeled examples set *U* and positive examples set *P*.

For information retrieval and text mining, a general definition of clustering is the following: given a large set of documents, automatically discover diverse subsets of documents that share a similar topic. Clustering provides unique ways of digesting large amounts of information. Clustering algorithms divide data into meaningful or useful groups, called clusters, such that the intra-cluster similarity is maximized and the inter-cluster similarity is minimized. These discovered clusters could be used to explain the characteristics of the underlying data distribution and thus serve as the foundation for various data mining and analysis techniques.

The standard clustering algorithms can be categorized into partitioning algorithms such as k-means and hierarchical algorithms such as Single-Link or Average-Link. Many variants of the k-means algorithm have been proposed for the purpose of text clustering. A recent study has compared partitioning and hierarchical methods of text clustering on a broad variety of test datasets. It concludes that k-means clearly outperforms the hierarchical methods with respect to clustering quality. A variant of k-means called bisecting k-means [10] is introduced, which yields even better performance. Bisecting k-means uses k-means to partition the dataset into two clusters. Then it keeps partitioning the currently largest cluster into two clusters, again using k-means, until a total number of k clusters has been discovered.

We propose a novel method for the first step as shown in fig. 1. First, set *RN* to null, and then run bisecting k-means clustering algorithm with the union of positive examples set and unlabeled examples set with parameter *k*. Last, if proportion of positive examples in each cluster is lower than the threshold that given, and then add this cluster to *RN*.

```
Algorithm: Exploiting Reliable Negative by Clustering
Input: P positive examples set
         U unlabeled examples set
         K number of cluster
         T threshold
Output: RN (reliable negatives set)
Steps: 1.RN ={};
         2.Clustering set E = P •U;
         3.run bisecting k-means with parameter k on E,
           and divide into E1, E2, … Ek, in each Ei(i = 1,
           2, …, k), the positive examples in it is Pi;
         4.for each Ei (i = 1, 2, … , k )
               if |Pi|/|Ei|<T then RN = RN •Ei.
```

Fig. 1. The algorithm of exploiting reliable negative by clustering

We use the CLUTO toolkit package [12] for clustering, which use bisecting k-means algorithm. The parameter *T* generally is small, usually set to zero, i.e. the cluster that has no positive examples can be used as reliable negative examples set. Yang, Y. suggests that the numbers of text clustering impacts the resulting difference in F_1 scores are almost negligible [11]. From our experiments in section 4, we also observed that the choice of *k* does not affect classification results much as long as it is not too small. So we set *k* as 20.

```
Algorithm: Iterative SVM
Input: P  positive examples set
        RN reliable negative examples set by step 1
        Q  the remaining unlabeled examples set, U -RN;
Output: The final classifier S;
Steps:
    1. Assigned the label 1 to each document in P;
    2. Assigned the label -1 to each document in RN;
    3. While(true)
    4.     Training a new SVM classifier Si with P and
RN;
    5.     Classify Q using S;
    6.     Let the set of documents in Q that are
           classified as negative be W;
    7.     If W ={} then break;
    8.     Else Q = Q - W; RN = RN •W;
    9.     End if
   10. End while
```

Fig. 2. The algorithm of iterative SVM

For step 2, we run SVM iteratively as shown in fig. 2. This method is similar to the step 2 of PEBL technique and Roc-SVM technique except that we do not use an additive classifier selection step. The basic idea is to use each iteration of SVM to exact more possible negative examples from Q ($U - RN$) and put them in RN. The iteration converges when no document in Q is classified as negative. Our technique does not select a good classifier from a set of classifiers built by SVM, and use the last SVM classifier at convergence. For Roc-SVM, the reason for selecting a classifier is that there is a danger in running SVM repetitively, since SVM is sensitive to noise. However, it is hard to catch the best classifier [6].

4 Experiments and Results

We now evaluate our proposed method with the Roc-SVM technique [7] and the PEBL technique [9]. We do not compare with the S-EM technique [8], because it uses the Naïve Bayesian method, which is a weaker classifier than the SVM, and our proposed technique is much more accurate than S-EM. Liu, B. et al. [6] have surveyed and compared these three methods, and our experiments on the dataset are with the same setting as [6] in order to allow comparison on the square.

4.1 Experiments Setup and Data Preprocess

We use Reuters-21578, the popular text collection in text classification experiment, which has 21578 documents collected from the Reuters newswire. Among 135 categories, only the most populous 10 are used. 9980 documents are selected to use in our experiment, as shown in Table 1. Each category is employed as the positive examples class, and the rest as the negative examples class. This gives us 10 datasets.

Table 1. The most popular 10 categories on Reuters-21578 and their quantity

Acq	Corn	Crude	Earn	Grain	Interest	Money-fx	Ship	Trade	Wheat
2369	**237**	**578**	**3964**	**582**	**478**	**717**	**286**	**486**	**283**

In data preprocessing, we use the Bow toolkit [13]. We applied stopword removal, but no feature selection or stemming was done. The tf-idf value is used in the feature vectors. For each dataset, 30% of the documents are randomly selected as test documents. The rest (70%) are used to create training sets as follows: γ percent of the documents from the positive examples class is first selected as the positive examples set P. The rest of the positive and negative documents are used as unlabeled examples set U. We range γ from 10%-90% to create a wide range of scenarios.

4.2 Evaluation Measures

In our experiments, we use the popular F_1 score on the positive examples class as the evaluation measure. F_1 score takes into account of both recall and precision. Precision, recall and F_1 defined as:

$$Precision = \frac{\#\ of\ \ correct\ \ positive\ \ predictions}{\#\ of\ \ positive\ \ predictions} \tag{2}$$

$$Recall = \frac{\#\ of\ \ correct\ \ positive\ \ predictions}{\#\ of\ \ positive\ \ examples} \tag{3}$$

$$F_1 = \frac{2 \times precision \times recall}{precision + recall} \tag{4}$$

For evaluating performance average across categories, there are two conventional methods, namely macro-average and micro-average. Macro-averaged performance scores are determined by first computing the performance measures per category and then averaging those to compute the global means. We use macro-averaging.

4.3 Experiment Results

In our experiments, we implemented the 1-DNF method used in PEBL and the Rocchio method in the Roc-SVM. We use the CLUTO toolkit package [12] for clustering, and set k=20. For SVM, we use the SVM^{light} system [14] with linear kernel, and do not tune the parameters. For Roc-SVM, we use α=16 and β=4 in formula (1).

We first compare the quantity of reliable negative examples produced by three methods. Table 2 shows the averaged quantity on the Reuters collection. The γ denotes the percent of the document from the positive examples class is selected as positive examples set P. For the PEBL, the quantity of initial negative examples is so small; by browsing the initial negative examples, we found these examples sometimes are short paper, and the quality is poor too. For the Rocchio method, the quantity of reliable negative examples is so big that near the two third of training data. For clustering method, the quantity is moderate, and sometimes balanced the training set.

Table 2. Averaged reliable negative quantity of three methods on the 10 Reuters collection

γ	PEBL	Rocchio	Clustering
10	**394.1**	**6253.9**	**3760.5**
20	**301.5**	**6894.7**	**3462.2**
30	**201.6**	**6642.3**	**3248.6**
40	**224.6**	**5845.0**	**3334.0**
50	**227.6**	**6793.0**	**3109.9**
60	**218.5**	**6779.1**	**2931.3**
70	**207.0**	**6802.5**	**2909.7**
80	**186.0**	**6816.3**	**2689.8**
90	**208.2**	**6863.6**	**2496.9**

Then we compare the F_1 score of our method with other two methods. The results of the PEBL method and the Rocchio method are extract from the experiment of Bing Liu et al. [6]. Fig.3 shows the macro-averaged F_1 score on the 10 Reuters datasets for each γ setting. When γ is smaller (<50), our method outperforms than other two. When γ is bigger, our method is as good as other methods. But there is still room for further improvement.

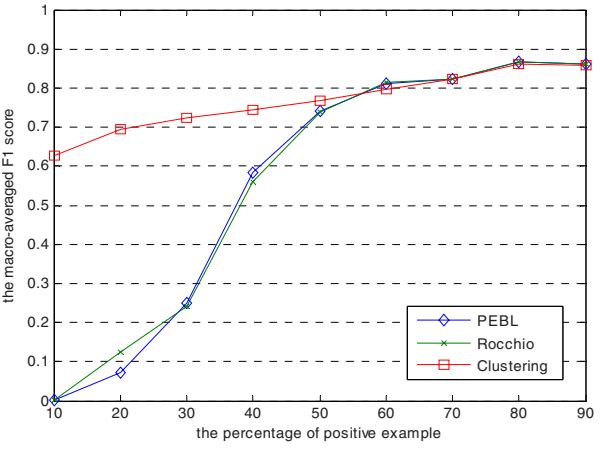

Fig. 3. The macro-averaged F1 scores on the Reuters 10 collection

The poor quality and quantity of the reliable negative examples by PEBL increase the number of the iterations of SVM, which ends up longer training time. The quantity of negative examples of Rocchio method is so big that biased the training set. Our proposed method produces the moderate quantity reliable negative examples.

5 Conclusion

In this paper, we discussed the two-step strategies for learning a classifier from positive examples and unlabeled examples data. The clustering method was added to the existing techniques. A comprehensive evaluation was conducted to compare their performances. Our method produces the moderate quantity reliable negative examples and sometimes balanced the training set. Our experiment shows that our method is efficient and effective. In particular, when positive examples are small, our method outperforms than other two; when γ is bigger, our method is as good as other methods.

Acknowledgments

The work was supported by the National Natural Science Foundation of China under Grant No. 60373099, the Project of the Jilin Province Science and Technology Development Plan under the Grant No.20070533, and the Science Foundation for Young Teachers of Northeast Normal University (No.20070602). We would like to thank the anonymous reviewers for their comments and suggestions.

References

1. Nigam, K., McCallum, A., Thrun, S., Mitchell, T.: Learning to Classify Text from Labeled and Unlabeled Documents. In: AAAI-98, pp. 792–799. AAAI Press, Menlo Park (1998)
2. Denis, F.: PAC Learning from Positive Statistical Queries. In: Richter, M.M., Smith, C.H., Wiehagen, R., Zeugmann, T. (eds.) ALT 1998. LNCS (LNAI), vol. 1501, pp. 112–126. Springer, Heidelberg (1998)

3. Letouzey, F., Denis, F., Gilleron, R.: Learning From Positive and Unlabeled Examples. In: Proceedings of 11th International Conference on Algorithmic Learning Theory (2000)
4. Denis, F., Gilleron, R., Tommasi, M.: Text Classification from Positive and Unlabeled Examples. In: Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (2002)
5. Denis, F., Gilleron, R., Laurent, A., Tommasi, M.: Text Classification and Co-Training from Positive and Unlabeled Examples. In: Proceedings of the ICML 2003 Workshop: The Continuum from Labeled to Unlabeled Data (2003)
6. Liu, B., Dai, Y., Li, L.X., Lee, W.S., Yu, P.: Building Text Classifiers Using Positive and Unlabeled Examples. In: Proceedings of the Third IEEE International Conference on Data Mining (2003)
7. Li, X.L., Liu, B.: Learning to Classify Text using Positive and Unlabeled Data. In: Proceedings of Eighteenth International Joint Conference on Artificial Intelligence (2003)
8. Liu, B., Lee, W.S., Yu, P., Li, X.L.: Partially Supervised Classification of Text Documents. In: Proc. 19th Intl. Conf. on Machine Learning (2002)
9. Yu, H., Han, J., Chang, K.C.C.: PEBL: Web Page Classification Without Negative Examples. J. IEEE Transactions on Knowledge and Data Engineering (Special Issue on Mining and Searching the Web) 16(1), 70–81 (2004)
10. Zhao, Y., Karypis, G.: Hierarchical Clustering Algorithms for Document Datasets. J. Data Mining and Knowledge Discovery 10(2), 141–168 (2005)
11. Yang, Y.: An Evaluation of Statistical Approaches to Text Categorization. J. of Information Retrieval 1(1/2), 67–88 (1999)
12. The CLUTO toolkit package, `http://glaros.dtc.umn.edu/gkhome/cluto/cluto/`download
13. Bow, A.: Toolkit for Statistical Language Modeling, Text Retrieval, Classification and Clustering, `http://www.cs.cmu.edu/~mccallum/bow/`
14. Joachims, T.: Making large-Scale SVM Learning Practical. In: Advances in Kernel Methods - Support Vector Learning, MIT Press, Cambridge (1999)