Reliable Negative Extracting Based on kNN for Learning from Positive and Unlabeled Examples

Bangzuo Zhang
College of Computer Science and Technology, Jilin University, Changchun, P. R. China
College of Computer, Northeast Normal University, Changchun, P. R. China
Email: zhangbz@nenu.edu.cn

Wanli Zuo
College of Computer Science and Technology, Jilin University, Changchun, P. R. China
Email: wanli@mail.jlu.edu.cn

Abstract—**Many real-world classification applications fall into the class of positive and unlabeled learning problems. The existing techniques almost all are based on the two-step strategy. This paper proposes a new reliable negative extracting algorithm for step 1. We adopt kNN algorithm to rank the similarity of unlabeled examples from the k nearest positive examples, and set a threshold to label some unlabeled examples that lower than it as the reliable negative examples rather than the common method to label positive examples. In step 2, we use iterative SVM technique to refine the finally classifier. Our proposed method is simplicity and efficiency and on some level independent to k. Experiments on the popular Reuter21578 collection show the effectiveness of our proposed technique.**

Index Terms—**Learning from Positive and Unlabeled examples, k Nearest Neighbor, Text Classification, Support Vector Machine, Information Retrieval**

I. INTRODUCTION

Traditional learning techniques typically require a large number of labeled examples to learn an accurate classifier. Thus, for binary problems, positive examples and negative examples are mandatory for machine learning and data mining algorithms such as decision tree and neural networks. This approach to building classifiers is called supervised learning. However, in many practical classification applications such as document retrieval and classification, positive information is readily available and unlabeled data can easily be collected, although it is possible to manually label some negative examples, it is labor–intensive and very time consuming. One way to reduce the amount of labeled training data needed is to develop classification algorithms that can learn from a set of labeled positive examples augmented with a set of unlabeled examples. That is give a set P of positive examples of a particular class and a set U of unlabeled examples, and then build a classifier using P and U to classify the data in U as well as future test data. A first example is web-page classification, suppose we want a program that classifies web sites as "interesting" for a web user. Positives examples are freely available: it is the set of web pages corresponding to web sites in his bookmarks. Moreover, unlabeled web pages are abundant, and easily available on the World Wide Web. Many real-world classification applications also can fall into this class problem. Such as, diagnosis of diseased: positive data are patients who have the disease, unlabeled data are all patients; marketing: positive data are clients who buy the product, unlabeled data are all clients in the database.

Denis originally proposes a framework for learning model from positive examples (POSEX for short) [1] based on the probably approximately correct model (PAC). The study concentrates on the computational complexity of learning and shows that function classes learnable under the statistical queries model are also learnable from positive and unlabeled examples. Liu et al [2] call this problem LPU (Learning form Positive and Unlabeled examples), while it is also called partially supervised classification [3], and PU learning problem [4]. Yu et al [5] introduce it as PEBL (Positive Example Based Learning). The key feature of this problem is that there is no labeled negative document, which makes traditional classification methods inapplicable, as they all need labeled examples of every class.

Recently, a few innovative techniques have been proposed to solve this problem. These algorithms include S-EM [2], Roc-SVM [3], PEBL [5] and NB [6]. One class of these techniques have focused on addressing the lack of labeled negative examples in the training examples, and based on a two-step strategy as follows:

Step 1: Extraction a set of negative examples called reliable negatives (*RN*) from the unlabeled examples U. In this step, S-EM uses a Spy technique, Roc-SVM uses the Rocchio algorithm, PEBL uses a technique called 1-DNF, and NB uses the Naive Bayes technique. The key requirement for this step is that the identified negative examples from the unlabeled examples must be reliable or pure, i.e., with no or very few positive examples.

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

© 2009 ACADEMY PUBLISHER

(Naive Bayes) classifier as the base classifier, while PEBL and Roc-SVM use Support Vector Machine (SVM). 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 in [2].

Other classes of methods for learning from positive and unlabeled examples are also presented. A NB based method (called PNB) [7] that tries to statistically remove the effect of positive data in the unlabeled set is proposed. The main shortcoming of this method is that it requires the user to give the positive class probability, which is hard for the user to provide in practice. It is also possible to discard the unlabeled examples and learn only from the positive examples. This was done in the one-class SVM [8], which tries to learn the support of the positive distribution. Some results [6] show that its performance is poorer than learning methods that take advantage of the unlabeled data.

kNN [9] stands for k-nearest neighbor classification, is a well-known statistical approach that has been intensively studied in pattern recognition. kNN is a type of instance-based learning, or lazy learning where the function is only approximated locally and all computation is deferred until classification. The kNN algorithm assigns each example to the majority class of its k closest neighbors where k is a parameter. For 1NN, the algorithm assigns each example to the class of its closest neighbor. The kNN algorithm is also an often-used method for the text categorization and has reported the best result in Reuter collection [9].

In this paper, we also follow the two-step strategy, and propose a novel method based on kNN algorithm for Step 1. We firstly use kNN algorithms to extract reliable negative and then construct an initial classifiers. We then use iterative SVM algorithm until its convergence. We carry out experiments in the popular Reuter21578 collection, and demonstrate the effectiveness of our proposed technique.

In this paper, we would like to first review the existing two-step LPU algorithms in Section 2, then propose a new reliable negative examples extracting method by kNN algorithm, and show its effectiveness experimentally on the Reuters 21578 collection in Section 4, finally make conclusion in Section 5.

II. RELATED WORKS

Given a set of training documents D, Each document is considered as an ordered list of words. We use $w_{d_i,k}$ to denote the word in position k of document d_i, where each word is from the vocabulary $V=<w_1,w_2, \cdots w_{|v|}>$. The vocabulary is the set of all the words considered for classification. For LPU, we only consider binary class classification, so a set of predefined class $C = \{c_0, c_1\}$, and we use c_0 for the positive class, while c_1 for negative class.

Traditional supervised learning and semi-supervised learning classification techniques require labeled training examples of all classes to build a classifier. They are thus not suitable for LPU problem. Recently, some LPU algorithms including S-EM [2], NB [6], Roc-SVM [3] and PEBL [5] are proposed, and they are all based on the two-step strategy. We firstly review the existing technique for step 1 in detail.

A. The Spy Technique in S-EM

The Spy technique in S-EM [2] first randomly selects a set S of positive documents from P and puts them in U. The default value is 10% (using 15% in [6]). The algorithm is given in Fig. 1. The spies behave identically to the unknown positive documents in P and hence allow to reliably inferring the behavior of the unknown positive documents in U. It then runs I-EM algorithm using the set P-S as positive and the set $U \cup S$ as negative (lines 3-7). I-EM basically runs NB twice. After I-EM completes, the resulting classifier uses the probabilities assigned to the documents in S to decide a probability threshold th to identify possible negative documents in U to produce the reliable negative examples set RN.

However, S-EM is not accurate because it uses naive Bayesian classifier as the underlying classifier in step 2. This algorithm performs stably when the positive set is very small. When the positive set is larger, it is worse than others.

B. The Naive Bayes Technique

The NB (Naive Bayes) technique is a popular method for text classification. Liu et al [6] first introduce it into LPU as a new method for step 1.

The NB classifier is constructed by using the training documents to estimate the probability of each class given the document feature values of a new instance. To perform classification, it computes the posterior probability, $Pr(c_j|d_i)$. Based on Bayesian probability and the multinomial model, it gives

1. $RN = \{\}$;
2. $S = Sample(P, s\%)$;
3. $Us = U \cup S$;
4. $Ps = P$-S;
5. Assign each document in Ps the class label 1;
6. Assign each document in Us the class label -1;
7. *I-EM*(Us, Ps); // This produces a NB classifier.
8. Classify each document in Us using the NB classifier;
9. Determine a probability threshold th using S;
10. For each document $d \in Us$
11. If its probability $Pr(1|d) < th$
12. Then $RN = RN \cup \{d\}$;
13. End If
14. End For

Figure 1. The spy technique in S-EM.

© 2009 ACADEMY PUBLISHER

$$\Pr(c_j) = \frac{\sum_{i=1}^{|D|} \Pr(c_j \mid d_i)}{|D|}, \quad (1)$$

To avoid zero probability estimates, some smoothing method is usually used. Liu et al [6] use the Lidstone smoothing as

$$\Pr(w_t \mid c_j) = \frac{\lambda + \sum_{i=1}^{|D|} N(w_t, d_i) \Pr(c_j \mid d_i)}{\lambda |V| + \sum_{s=1}^{|V|} \sum_{i=1}^{|D|} N(w_s, d_i) \Pr(c_j \mid d_i)}, \quad (2)$$

where λ is the smoothing factor, $N(w_t,d_i)$is number of times that word w_t occurs in document d_i and $pr(c_j|d_i) \in \{0, 1\}$ depending on the class of the document.

Assuming that the probabilities of the words are independent given the class, the NB classifier has been defined as equation (3)

$$\Pr(c_j \mid d_i) = \frac{\Pr(c_j) \prod_{k=1}^{|d_i|} \Pr(w_{d_i,k} \mid c_j)}{\sum_{r=1}^{|C|} \Pr(c_r) \prod_{k=1}^{|d_i|} \Pr(w_{d_i,k} \mid c_r)}. \quad (3)$$

In classifying a document d_i, the class with the highest $Pr(c_j|d_i)$ is assigned as the class of the document. The method of extracting a set *RN* of reliable negative documents from the unlabeled examples set *U* is done briefly as Fig. 2. Despite the fact that the assumption of conditional independence is generally not true for word appearance in documents, the naive bayes classifier is surprisingly effective.

C. The Rocchio Technique

The Roc-SVM algorithm [3] uses the Rocchio method to identify a set *RN* from *U*, which is a classic method for document routing or filtering in information retrieval. Building a Rocchio classifier is achieved by constructing a prototype vector for each class *j* following equation (4)

$$cj = \alpha \frac{1}{|cj|} \sum_{\vec{d} \in Cj} \frac{\vec{d}}{\|\vec{d}\|} - \beta \frac{1}{|D - cj|} \sum_{\vec{d} \in D - Cj} \frac{\vec{d}}{\|\vec{d}\|}. \quad (4)$$

α and β are parameters that adjust the relative impact of relevant and irrelevant training examples. Generally use $\alpha = 16$ and $\beta = 4$.

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*.

The algorithm that uses Rocchio to identify a set *RN* from *U* is the same as that in Fig. 2 except that it replaces NB with Rocchio. Rocchio performs well consistently under a variety of conditions.

1. Assign label 1 to each document in *P*;
2. Assign label –1 to each document in *U*;
3. Build a NB classifier using *P* and *U*;
4. Use the classifier to classify *U*. Those documents in *U* that are classified as negative form the reliable negative set *RN*.

Figure 2. The method of extracting RN using NB.

D. The 1-DNF Technique for PEBL

Yu et al [5] proposes the PEBL framework for web page classification, which uses mapping-convergence algorithm. In the mapping stage, they extract reliable negative from the unlabeled data by the 1-DNF method.

The 1-DNF algorithm is given in Fig. 3. It firstly builds a disjunction list of positive feature set *PF* which contains words that occur in the positive examples set *P* more frequently than in the unlabeled examples set *U* (line 2-6). Then it tries to filter out possible positive documents from *U* (line 8-12). 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.

PEBL is not robust because it performs well in certain situations and fails badly in others. PEBL is sensitive to the number of positive examples. When the positive data is small, the results are often very poor.

E. Techniques in Step 2

There are four techniques for the second step:

1. Running SVM only once using sets *P* and *RN* after step 1. This method is seldom to use.
2. Running EM. This method is used in S-EM [2].
3. Running SVM iteratively. This method is used in PEBL [5].
4. Running SVM iteratively and then selecting a final classifier. This method is used in Roc-SVM [3].

The Expectation-Maximization (EM) algorithm is a popular iterative algorithm for maximum likelihood estimation in problems with missing data. The EM algorithm consists of two steps, the Expectation step, and the Maximization step. The Expectation step basically fills in the missing data. It produces and revises the probabilistic labels of the documents in $Q = U - RN$. The parameters are estimated in the Maximization step after the missing data are filled. This leads to the next iteration of the algorithm. EM converges when its parameters stabilize. The EM algorithm iteratively runs NB to revise the probabilistic label of each document in set *Q*.

1. $PF = \{\}$
2. For $i = 1$ to n
3. If $(freq(w_i, P)/|P| > freq(w_i, U)/|U|)$
4. Then $PF = PF \cup \{w_i\}$
5. End if
6. End for
7. $RN = U$;
8. For each document $d \in U$
9. If $\exists w_i \, freq(w_i, d) > 0$ and $w_i \in PF$
10. Then $RN = RN - \{d\}$
11. End if
12. End for

Figure 3. The 1-DNF technique in PEBL.

© 2009 ACADEMY PUBLISHER

SVM is an effective learning algorithm for text classification. The iterative SVM algorithm results in the best performance (see Section III.C for details). The reason for selecting a classifier is that there is a danger in running SVM repetitively. Since SVM is sensitive to noise, if some iteration of SVM extracts many positive documents from *Q* and put them in *RN*, then the last SVM classifier will be poor. However, it is hard to catch the best classifier.

Liu et al [6] perform an evaluation of all 16 possible combinations of methods for step 1 and step 2 on the Reuters 21578 and the 20 Newsgroup corpuses.

III. THE PROPOSED TECHNIQUES

In this section, we proposed a novel technique for LPU problem based on the two-step strategy. First, introduce a new reliable negative examples extracting method based on kNN algorithm. Although the kNN algorithm can't be applied directly to LPU problem, we use it as a ranking process, and set a threshold to label the reliable negative examples set *RN*. In step 2, we use the SVM iteratively to produce the final classifier.

A. Introduction to kNN Algorithm

The k-nearest neighbor algorithm [10] is amongst the simplest of all machine-learning algorithms. kNN algorithm requires no explicit training and can use the unprocessed training set directly in classification. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common amongst its *k* nearest neighbors. The parameter *k* in kNN is often chosen based on experience or knowledge about the classification problem at hand. And *k* is a positive integer, typically small. If *k* equals 1, then the object is simply assigned to the class of its nearest neighbor. In binary (two class) classification problems, it is desirable for *k* to be odd to make ties less likely. The same method can be used for regression, by simply assigning the property value for the object to be the average of the values of its *k* nearest neighbors. It can be useful to weight the contributions of the neighbors, so that the nearer neighbors contribute more to the average than the more distant ones.

In kNN classification, user need not perform any estimation of parameters as using Rocchio (centroids) classification or in Naive Bayes (priors and conditional probabilities). kNN simply memorizes all examples in the training set and then compares the test examples to them. For this reason, kNN is also called memory-based learning or instance-based learning.

The neighbors are taken from a set of objects for which the correct classification (or, in the case of regression, the value of the property) is known. This can be thought of as the training set for the algorithm, though no explicit training step is required. In order to identify neighbors, the objects are represented by position vectors in a multidimensional feature space. It is usual to use the Euclidean distance, though other distance measures, such as the Manhattan distance could in principle be used instead. The k-nearest neighbor algorithm is sensitive to the local structure of the data.

The nearest-neighbor rule is a sub-optimal procedure. Its use will usually lead to an error rate greater than the minimum possible, i.e. the Bayes error rate. However, with an unlimited number of prototypes the error rate is never worse than twice the Bayes error rate [10]. kNN's effectiveness is close to that of the most accurate learning methods in lots of applications.

The kNN algorithm is also an often-used method for the text categorization [9]. Given a test document, the system finds the *k* nearest neighbors among the training documents, and uses the categories of the *k* neighbors to weight the category candidates. The similarity score of each neighbor document to the test document is used as the weight of the categories of the neighbor document. If several of the *k* nearest neighbors shares a category, then the per-neighbor weights of that category are added together, and the resulting weighted sum is used as the likelihood score of that category with respect to the test document. By sorting the scores of candidate categories, a ranked list is obtained for the test document. By set a threshold on these scores, binary category assignments are obtained. The decision rule [9] in kNN has been written as equation (5)

$$y(d, c_j) = \sum_{d_i \in kNN} sim(d, d_i) y(d_i, c_j) - b_j \ , \quad (5)$$

where $y(d, c_j)$ is the classification for document d with respect to category c_j; $sim(d, d_i)$ is the similarity between the test document d and the training document d_i; and b_j is the category-specific threshold for the binary decisions.

For the parameter *k* in kNN, Y. Yang [9] tests the values of 30, 45 and 65, and suggests that the resulting difference in the F1 scores of kNN are almost negligible. So, in [11] they set *k* as 45.

B. The RN Extracting Technique using kNN

The kNN algorithm can't be applied directly to LPU problem. However, there is a possibility to employ a process of ranking [12, 13]. The unlabeled examples are ranked according to their similarity to the training samples. When the distances of unlabeled examples from *k* nearest positive examples are computed, the resulting values can be used for sorting the classified examples, nearer unlabeled instances take positions ahead of the ones that are further away. J., Hroza et al [12, 13] then decide what is the 'true' similarity, how many unlabeled examples they are willing to accept, what degree of precision is acceptable, and what recall is still satisfactory. According to priorities assigned to the parameters of the kNN Ranking algorithm, they label the first *r* vectors as positive examples.

J., Hroza et al do not give an operable method for how to decide the appropriate value of *r*. When user is interested only in a small part of the most relevant documents, this method can get very high precision, but the recall value is very small, so the smaller F1 score. However, it is hard to determine the value of *r*.

© 2009 ACADEMY PUBLISHER

We follow the thought of rank, and reverse the method to exact the reliable negative examples rather than to label the positive examples. That is, For LPU problem, we set a predefined threshold *T*, if the similarity resulting value of an unlabeled example is lower than *T*, and then label it as reliable negative example. We consider that unlabeled examples are very large, so not need to set the *T* elaborately. When we exact the pure reliable negative examples set, then we can use some methods to refine the classifier in step 2. According to our method, the decision rule can be rewritten as equation (6)

$$w(d) = \sum_{d_i \in kNN} sim(d, d_i) - T \,. \tag{6}$$

When kNN is applied to text examples, we tokenize all documents by a vector with TFIDF weight following the traditional Information Retrieval (IR) approach. Assuming the term vectors are normalized, cosine function is a commonly used similarity measure for two documents as equation (7)

$$sim(d_i, d_j) = \sum_{m=1}^{|v|} w_{im} \cdot w_{jm} \,. \tag{7}$$

For the parameter *k* in kNN, J., Hroza et al [12] test *k* from 1 to 5, and when *k* is 5 get the best result on Reuters 10 dataset. When consider the different word representations and stop-word number, they get the different conclusion [12, 13].

Our proposed reliable negative extracting algorithm using kNN is shown in Fig. 4.

C. The Iterative SVM technique

Support Vector Machines (SVM) is a relatively new learning approach introduced by Vapnik in 1995 for solving two-class pattern recognition problems [14]. It is based on the Structural Risk Minimization principle for which error-bound analysis has been theoretically motivated. The idea of structural risk minimization is to find a hypothesis for which can guarantee the lowest true error.

SVM are very universal learners in text classification. In their basic form, SVM learn linear threshold function. Nevertheless, by a simple "plug-in" of an appropriate kernel function, they can be used to learn polynomial classifiers, radial basic function (RBF) networks, and three-layer sigmoid neural nets. One remarkable property of SVM is that their ability to learn can be independent of the dimensionality of the feature space.

Considering a binary classification task with data points x_i (i = 1, … , n), having corresponding labels y_i = +1 or -1 and let the decision function be

$$f(x) = sign(w \bullet x + b) \,. \tag{8}$$

The problem of finding the hyper plane can be stated as the following optimization problem

$$\begin{aligned} &Minimize: \frac{1}{2} w^T w \\ &Subjectto: y_i(w^T x_i + b) \geq 1, i = 1,2,\cdots,n \end{aligned} \,. \tag{9}$$

To deal with cases where there may be no separating hyper plane due to noisy labels of both positive and negative training examples, the soft margin SVM is proposed, which is formulated as

$$\begin{aligned} &Minimize: \frac{1}{2} w^T w + C\sum_{i=1}^{n} \xi_i \\ &Subjectto: y_i(w^T x_i + b) \geq 1 - \xi_i, i = 1,2,\cdots,n \end{aligned} \,. \tag{10}$$

where $C \geqslant 0$ is a parameter that controls the amount of training errors allowed.

Joachims [15] firstly introduces support vector machines for text categorization. The experimental results show that SVM consistently achieve good performance on text categorization tasks, outperforming existing methods substantially and significantly. From theoretical and empirical evidence, he concludes that SVM acknowledge the particular properties of text: (a) high dimensional feature spaces, (b) few irrelevant features (dense concept vector) and (c) sparse instance vector.

With its ability to generalize well in high dimensional feature spaces, SVM eliminates the need for feature selection, making the application of text categorization considerably easier. Another advantage of SVM over the conventional methods is their robustness. Furthermore, SVM do not require any parameter tuning, since they can find good parameter settings automatically. All this makes SVM a very promising and easy-to-use method for learning text classifiers from examples.

For step 2, we run SVM iteratively as shown in Fig. 5. 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. Our technique does not select a good classifier from a set of classifiers built by SVM, and use the last SVM classifier at convergence. The basic idea is to use each iterations of SVM to exact

Algorithm: Reliable negative extracting using kNN
Input: *P* positive examples set
U unlabeled examples set
K the number of nearest neighbors
T threshold
Output: RN reliable negative examples set;
Steps:
1. $RN = \{\}$
2. For each unlabeled examples u_i
3. For each positive examples v_j
4. Computing the similarity $sim(u_i, v_j)$
5. End For
6. Select *k* nearest neighbors v_j (j=1,…,k)
7. Compute the result value $w(u_i)$ according to equation 6
8. If $w(u_i)$<0;
9. Then $RN = RN \cup u_i$
10. End If
11. End For

Figure 4. Reliable negative extracting using kNN.

© 2009 ACADEMY PUBLISHER

Algorithm: Iterative SVM
Input: *P* positive examples set
RN reliable negative set produced by step 1
Q the remaining unlabeled set, i.e. *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 *S* with *P* and *R*N;
5. Classify *Q* using *S*;
6. Let the set of documents in *Q* that are classified as negative be *W*;
7. If $W \neq \{\}$
8. Then $Q = Q\text{-}W$; $RN = RN \cup W$;
9. End If
10. End While

Figure 5. The algorithm of iterative SVM

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.

H. Yu et al [5] analysis that as long as the initial positive and negative examples is strong, the iterative SVM can converge into the unbiased negatives through the iterations regardless of the quality of the initial mapping. The poor quality of the initial mapping would increase the number of the iterations in the algorithm, which ends up longer training time, but the final accuracy would be the same. Our experiments also show that classification accuracy converges into the traditional SVM trained from the labeled examples no matter how bad the initial mapping is.

IV. Experiment

We now evaluate our proposed technique, and compare with the originally LPU algorithms. That is S-EM [2], Roc-SVM [3], PEBL [5] and NB [6].

A. Experiments Setup and Data Preprocess

We use Reuters-21578 [16], 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. Each category is employed as the positive examples class, and the rest as the negative examples class. This gives us 10 datasets. Table 1 gives the number of documents in each of the ten topic categories.

In data preprocessing, we use the Bow toolkit [17]. We applied stop-word removal, and the stop-list is the SMART system's list of 524 common words, not consider the number of stop-words as that J., Hroza et al [12, 13] do. No feature selection or stemming was done. The TFIDF 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.

TABLE I.
THE MOST POPULAR 10 CATEGORIES IN REUTERS-21578

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

B. Evaluation Measures

In our experiments, we use the popular F1 score on the positive examples class as the evaluation measure. F1 score takes into account of both recall and precision. The F1 measure is often used as an optimization criterion in threshold tuning for binary decisions. Its score is maximized when the values of recall and precision are equal or close; otherwise, the smaller of recall and precision dominates the value of F1. Precision, recall and F1 defined as:

$$Precision = \frac{\#\ of\ correct\ positive\ predictions}{\#\ of\ positive\ predictions}, \quad (11)$$

$$Recall = \frac{\#\ of\ correct\ positive\ prediction\ s}{\#\ of\ positive\ examples}, \quad (12)$$

$$F_1 = \frac{2 \times precision \times recall}{precision + recall}. \quad (13)$$

For evaluating performance average across categories, we use macro averaging. Macro averaging performance scores are determined by first computing the performance measures per category and then averaging those to compute the global means.

C. Experiment Results

We implemented our proposed algorithm. For SVM, we use the SVM^{light} system [18] with linear kernel, and do not tune the parameters. The results of PEBL, S-EM, ROC-SVM, and the NB method are extracted from the experiment of Liu et al [6]; Noted that they are all use the

© 2009 ACADEMY PUBLISHER

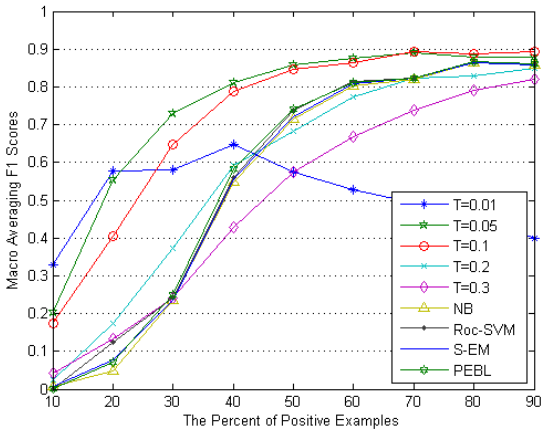

Figure 6. The Macro averaging F1 scores of our proposed method with different T values and compare with the other four LPU algorithms.

iterative SVM technique in step 2 and comparable with our proposed method.

First, based on the work of J., Hroza et al [12, 13], we set k to 5, and test the T value range from 0.01 to 0.3, the macro-averaging $F1$ score on the 10 Reuters datasets for each γ setting are shown in Fig. 6. We compare the macro averaging F1 score with the other LPU algorithm. When T is 0.05 and 0.01, our proposed method outperforms others, especially when the percent of positive is small. We find that when T equals 0.05 almost get the best result, so we set it as the T value in the next experiments. And we can observe that the value of T has the significant impact on the F1 score. So, how to tune the value is one of the important future works.

Second, we set T to 0.05, and test different k values for kNN. We not only test 1, 2, and 5 value that has been used in [12, 13], but also 45 that used in [9, 11], and we also test the values of 10, 20, and 30. The Macro averaging F1 scores of our proposed method with different k values are shown in Fig. 7. Our experiments

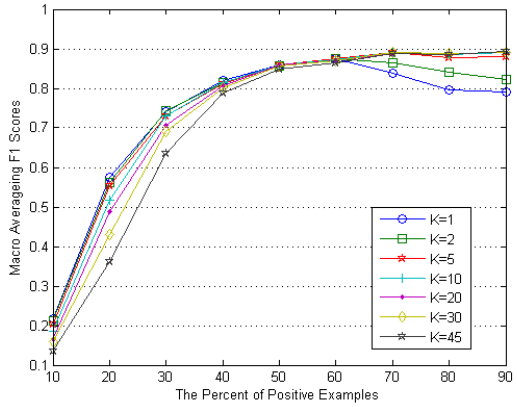

Figure 7. The Macro averaging F1 scores of our proposed method with different k values

confirmed Y. Yang's conclusion [11] that the impact of parameter k for resulting difference in the F1 scores of kNN are almost negligible. So, it not needs to elaborately tune the K value.

The result is exciting. Our proposed method performs well and on some level independent to k. Experiments show the effectiveness of our proposed technique.

V. CONCLUSION

Many real-world classification applications fall into the class of positive and unlabeled learning problems. In this paper, we propose a new reliable negative example set extracting algorithm that use kNN for solving LPU problem based on the two-step strategy. We adopt kNN algorithm to rank similarity of unlabeled examples from the k nearest positive examples, and then set a threshold to label some unlabeled examples that lower than it as the reliable negative examples, and contrary to J., Hroza et al's work that labels positive examples. For step 2, we use iterative SVM technique to refine the classifier. Experiments on the popular Reuter21578 collection show the effectiveness of our proposed technique. Our proposed technique is simplicity and efficiency and on some level independent to k.

Besides tuning the threshold T for rank learning with kNN algorithm, larger testing with more real data could bring more accurate answers, which also is the aim of the future works.

ACKNOWLEDGMENT

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

REFERENCES

[1] F. Denis, "PAC Learning from Positive Statistical Queries", *Proc. of Workshop on Algorithmic Learning Theory*, Springer, Heidelberg, 1998, pp. 112-126.

[2] B. Liu, Y. Dai, X.L. Li, W.S. Lee, and Philip Y., "Building Text Classifiers Using Positive and Unlabeled Examples", *ICDM-03*, Melbourne, Florida, November 2003, pp. 19-22.

[3] B. Liu, W.S. Lee, P.S. Yu, and X.L. Li, "Partially Supervised Classification of Text Documents", *Proceedings of the Nineteenth International Conference on Machine Learning (ICML-2002)*, Sydney, July 2002, pp. 387-394.

[4] X.L. Li and B. Liu, "Learning to Classify Documents with Only Positive Training Set", *ECML 2007*, LNAI 4701, 2007, pp.201-213.

[5] H. Yu, J. Han, and Chang K.C.-C., "PEBL: Positive Example Based Learning for Web Page Classification Using SVM", *Proc. Eighth Int'l Conf. Knowledge Discovery and Data Mining (KDD'02)*, ACM Press, New York, 2002, pp. 239-248.

[6] B. Liu, Y. Dai, X.L. Li, W. S. Lee, and Philip Y., "Building Text Classifiers Using Positive and Unlabeled

© 2009 ACADEMY PUBLISHER

Examples", *Proceedings of the Third IEEE International Conference on Data Mining (ICDM-03)*, Melbourne, Florida, November 2003, pp. 19-22.
[7] F., Denis, R., Gilleron and M. Tommasi, "Text classification from positive and unlabeled examples", *IPMU*, 2002.
[8] L., Manevitz and M., Yousef, "One-class SVMs for document classification", *Journal of Machine Learning Research*, vol. 2, 2001, pp.139-154.
[9] Y., Yang and X. Liu, "A Re-Examination of Text Categorization Methods", *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, August 15-19, 1999, Berkeley, CA, USA, pp.42-49.
[10] Duda, R. O., Hart, P. E., and Stock, D. G., *Pattern Classification*, Second Edition, John Wiley& Sons, 2001
[11] Y. Yang, "An evaluation of statistical approaches to text categorization", *Journal of Information Retrieval*, 1999 volume 1, pp.67-88.
[12] J., Hroza and J. Žižka, B. Pouliquen, C. Ignat and R. Steinberger, "Mining Relevant Text Documents Using Ranking-Based k-NN Algorithms Trained by Only Positive Examples", *Proceedings of the Fourth Czech-Slovak Conference Knowledge-2005*, February 9-11, 2005, Stará Lesná, Slovak Republic, pp. 29-40.
[13] J., Hroza, J. Žižka, B. Pouliquen, C. Ignat and R. Steinberger, "The Selection of Electronic Text Documents Supported by Only Positive Examples", *JADT 2006*, Besancon, France, pp. 1001-1010.
[14] V. Vapnic, *the Nature of Statistical Learning Theory*, Springer, New York, 1995.
[15] T. Joachims, "Text Categorization with Support Vector Machines: Learning with Many Relevant Features", *European Conference on Machine Learning (ECML)*, 1998.
[16] Reuters-21578 Text Categorization Collection, http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html.
[17] Bow: A Toolkit for Statistical Language Modeling, Text Retrieval, Classification and Clustering, http://www.cs.cmu.edu/~mccallum/bow/.
[18] T. Joachims, "Making large-Scale SVM Learning Practical", *Advances in Kernel Methods - Support Vector Learning*, MIT-Press, 1999

Bangzuo Zhang, born in Langzhong City, Sichuan Province, P.R.China, on Feb. 27, 1971. Received Bachelor of Science in computer science education from Northeast Normal University, China in 1995 and Master of Engineer in computer application technique from Jilin University, China in 2002. From Sep. 2003, Ph D. candidate in computer science and technology from Jilin University, China.

He has been a faculty number from July 1995. Currently, he is a Lecturer in College of Computer, Northeast Normal University, China. He has joined and accomplished 3 national and provincial research programs, such as "Research on the Semi-supervised Text Mining and Application". He has also published 6 papers in International conferences/journals, such as "A Novel Reliable Negative Method Based Clustering For Learning from Positive and Unlabeled Examples" in Lecture Notes in Computer Science. His major research interests include database and intelligent network, web intelligence.

Mr. Zhang has received the First Class Educational Achievement Award from Higher Educational Committee of Jilin Province, China in 2000.

Wanli Zuo, born in Jilin City, Jilin Province, P.R.China, on Dec. 6, 1957. Received Bachelor of Engineering, Master of Science, and Ph D. from Jilin University, P.R.China in 1982, 1985, and 2005 respectively.

He has been working in Jilin University since 1985. From July 1996 to July 1997, he conducted collaborative research in Louisiana Stare University, US, as a senior visiting scholar. He has accomplished 5 national and provincial research programs, such as "Object-oriented Active database based on Petri nets". He has also published more than 60 papers in International conferences/journals and Chinese Journals, such as "Relationship Graph and Termination Analysis of Active Rules in Database Systems" in Chinese Journal of Software. He has also published 4 books, such as "A Course of Operating Systems" by Higher Educational Press of P. R. China in 2004. His major research interests include database, web intelligence, and search engines.

Dr. Zuo is currently senior member of Computer Federation of China, council of System Software Association of China. Received 5 awards from Educational Department of China, such as the Second Class National Educational Achievement Award in 1996.

© 2009 ACADEMY PUBLISHER