Building High-Performance Classifiers Using Positive and Unlabeled Examples for Text Classification*

Ting Ke[1], Bing Yang[1], Ling Zhen[1], Junyan Tan[1], Yi Li[2], and Ling Jing[1,**]

[1] Department of Applied Mathematics, College of Science, China Agricultural University,
100083, Beijing, P.R. China
{kk.ting,zhenling38}@163.com, yangbing93@sohu.com,
tanjunyan0@126.com, jingling@cau.edu.cn
[2] Department of Mathematics, School of Science, Beijing University of Posts
and Telecommunications, 100876, Beijing, P.R. China
liyi0209@sina.com

Abstract. This paper studies the problem of building text classifiers using only positive and unlabeled examples. At present, many techniques for solving this problem were proposed, such as Biased-SVM which is the existing popular method and its classification performance is better than most of two-step techniques. In this paper, an improved iterative classification approach is proposed which is the extension of Biased-SVM. The first iteration of our developed approach is Biased-SVM and the next iterations are to identify confident positive examples from the unlabeled examples. Then an extra penalty factor is given to weight these confident positive examples error. Experiments show that it is effective for text classification and outperforms the Biased-SVM and other two step techniques.

Keywords: text classification, PU learning, SVM.

1 Introduction

With an increasing number of documents on the web, it is very important to build a text classifier which can identify a class of documents. In traditional classification, the user first collects a set of training examples, which are labeled with pre-defined classes. A classification algorithm is then applied to the training data to build a classifier. This approach to building classifiers is called supervised learning [3, 14]. In addition, Semi-supervised text classification makes use of unlabeled data to alleviate the intensive effort of manually labeling. Compared with semi-supervised text classification [12, 13], no pre-given negative training examples is required and unlabeled examples contain positive examples and negative examples. For example in practice, users may mark their favorite Web pages, but they are usually unwilling to

* Supported by the National Natural Science Foundation of China (10971223, 11071252), Chinese Universities Scientific Fund(2011JS039, 2012Y130).

** Corresponding author.

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 187–195, 2012.

mark boring pages. Because of its great application, we concentrate on PU (positive data and unlabeled examples) learning which is regarded as a special form of semi-supervised text classification in this paper.

Various approaches have been suggested in the literature to solve PU learning. In the first approach, the dataset consists of only labeled positive examples. One-class SVM [4] is one such approach and it estimate the distribution of positive examples without using unlabeled examples. In the second approach, two-step strategies [2, 7, 8, 9, 10, 15], step one: extract reliable negative or positive examples from unlabeled data to enlarge the original training set and step two: train text classifiers using original positive examples and reliable negative or positive examples (RN or RP). At present, most methods adopt this approach. In the third approach, one-step method is proposed to solve the problem which is the most related work with ours [6, 11]. For example, Biased-SVM is built by giving appropriate weights to the positive examples P and unlabeled examples U which is regarded as negative examples with noise respectively. It was shown in [10] that if the sample size is large enough, minimizing the number of unlabeled examples classified as positive while constraining the positive examples to be correctly classified will give a good classifier. Therefore, experimental results indicate that the performance is better than most of two-step strategies.

However, it is not reasonable to give equally weights to all unlabeled examples error because it also contains positive examples in U. In fact, the reliability of each example in U is different. i.e., the confidence of positive (CP) from U is lower than P, but higher than negative examples (N). Therefore, a novel iterative method is proposed in this paper which regards Biased-SVM as first iteration step and evaluates different weights to different example in U at next iteration steps for text classification. Experimental results indicate that the proposed method outperforms traditional methods.

2 Related Work

In this Section, we briefly review previous work related to this paper.

2.1 Support Vector Machine

The SVM is a promising classification technique proposed by Vapnik and his group at AT&T Bell Laboratories [1]. Different from classical methods that mainly minimize the empirical training error, SVM seeks an optimal separating hyper-plane that maximizes the margin between two classes after mapping the data into a feature space. Consider a binary classifier, which uses a hyper-plane to separate two classes based on given training examples $\{x_i, y_i\}$ for $i = 1, \cdots, l$, where x_i is a vector in the inpue space R^n and y_i denotes the class label taking a value $+1$ or -1 . The SVM solution is obtained through maximizing the margin between the separating hyper-plane and the data, where the margin is defined as $2/\|w\|$. The optimal hyper-plane is required to satisfy the following constrained minimization

$$\min_{w,b} \frac{1}{2}||w||^2$$

$$\text{s.t. } y_i (w \cdot x_i + b) \geq 1, \ i = 1, \ldots, l \ . \tag{1}$$

It then searches for a linear decision function

$$f(x) = w \cdot x + b \ . \tag{2}$$

in the input space S. For any test instance x, if $f(x) > 0$, it is classified into the positive class; otherwise, it belongs to the negative class.

For the linearly non-separable case, the minimization problem needs to be modified to allow the misclassification data points. This modification results in a soft margin classifier that allows but penalizes errors by introducing a new set of variables $\xi_i, \ i = 1, \cdots, l$.

$$\min_{w,b,\xi} \frac{1}{2}||w||^2 + C\sum_{i=1}^{l} \xi_i$$

$$\text{s.t. } y_i(w \cdot x_i + b) \geq 1 - \xi_i \,, i = 1, \ldots, l \,; \tag{3}$$

$$\xi_i \geq 0, i = 1, \ldots, l \,.$$

2.2 Biased-SVM

For PU learning, Liu et al propose a one-step method called Biased-SVM [11]. Biased-SVM takes unlabeled example as negative examples with noise. And then the classifier is built by giving appropriate weights to the positive examples error and unlabeled examples error respectively. Let P is positive examples set and U is unlabeled examples set. m and n is the number of positive and unlabeled examples. For training examples $x_i \in P, \ y_i = 1$; $x_i \in U$, $y_i = -1$. The following problem need to be solved.

$$\min_{w,b,\xi} \frac{1}{2}||w||^2 + C_+ \sum_{i=1}^{m} \xi_i + C_- \sum_{i=m+1}^{m+n} \xi_i$$

$$\text{s.t. } y_i(w \cdot x_i + b) \geq 1 - \xi_i, \ i = 1, \ldots, m + n \,; \tag{4}$$

$$\xi_i \geq 0, \ i = 1, \ldots, m + n \,.$$

Where C_+ and C_- represent the penalty factors of misclassification for positive and unlabeled example sets respectively. $\xi_i, \ i = 1, \cdots, m + n$ is penalizing variables. Experiment results indicate that the performance is better than most of two-step strategies. Nevertheless it is not reasonable to give equally weights to all unlabeled examples because it also contains positive examples in U.

3 Our Method

In this section, we improve on Biased-SVM based on its shortcoming for PU learning. We first present the formulation of our method. Then, an efficient learning algorithm will be introduced.

3.1 An Extension Algorithm Based on Biased-SVM (EB-SVM)

The purpose of PU learning is to find a classifier which identify class label for a given example x. Suppose the training examples include a small number of positive examples (P), and a large number of unlabeled examples (U). Unlabeled examples are mixed with other positive examples and negative examples (N). Assume that the fraction of positive examples in U is $\delta = \mathrm{r/n}$, where r is the number of positive examples approximately in U.

For training examples, our goal is to extract as many as real positive examples from U. i.e., if only we identify enough true positive examples from U, the most of rest examples in U are negative examples. Therefore, all the value of precision and recall are high. Based on above thought, we try to improve on Biased-SVM by giving another penalty factor to the examples which are identified positive examples from U. These positive examples are called confident positive examples. Now, we can minimize the following formulation which uses three penalty factors $\mathrm{C_p}$, $\mathrm{C_{rp}}$ and $\mathrm{C_n}$ to weight positive errors, confident positive errors and negative errors.

$$\min_{\mathrm{d}} \min_{\mathrm{w,b,\xi}} \frac{1}{2}\|\mathrm{w}\|^2 + \mathrm{c_p}\sum_{\mathrm{i=1}}^{\mathrm{m}} \xi_{\mathrm{i}} + \mathrm{c_{rp}} \sum_{\mathrm{i=m+1}}^{\mathrm{m+n}} \mathrm{d_i}\xi_{\mathrm{i}} + \mathrm{c_n} \sum_{\mathrm{i=m+1}}^{\mathrm{m+n}} (1-\mathrm{d_i})\xi_{\mathrm{i}}$$

$$\begin{aligned} \mathrm{s.t.}\ & \mathrm{d_i}\,[(\mathrm{w}\cdot\mathrm{x_i}+\mathrm{b}) \geq 1-\xi_{\mathrm{i}}]\,,\ \mathrm{i}=1,\cdots \mathrm{m}+\mathrm{n}\,; \\ & (1-\mathrm{d_i})\,[-(\mathrm{w}\cdot\mathrm{x_i}+\mathrm{b}) \geq 1-\xi_{\mathrm{i}}]\,,\ \mathrm{i}=1,\cdots \mathrm{m}+\mathrm{n}\,; \\ & \textstyle\sum_{\mathrm{i=1}}^{\mathrm{m+n}} \mathrm{d_i} \leq \mathrm{m}+\mathrm{r}\,; \\ & \mathrm{d_i}=1,\ \ \mathrm{i}=1,\ldots \mathrm{m}\,; \\ & \mathrm{d_i}=\{0{,}1\}\,,\ \ \mathrm{i}=\mathrm{m}+1,\ldots,\mathrm{m}+\mathrm{n}\ \ ; \\ & \xi_{\mathrm{i}} \geq 0,\ \ \mathrm{i}=1,\cdots,\mathrm{m}+\mathrm{n}\,; \\ & \mathrm{d}=\{\mathrm{d_1},\ \mathrm{d_2},\cdots,\ \mathrm{d_{m+n}}\}\,. \end{aligned} \tag{5}$$

Where d is the balance constraint which avoids the trivial solution that assigns all the unlabeled instances to the same class. $\xi_{\mathrm{i}},\ \mathrm{i}=1,\cdots,\mathrm{m}+\mathrm{n}$ is slack variables which allows the misclassification of some training examples. For U, $\mathrm{x_i},\ \mathrm{i}=\mathrm{m}+1,\cdots,\mathrm{m}+\mathrm{n}$ are confident positive examples if $\mathrm{d_i}=1$, otherwise they are negative examples. We can vary $\mathrm{C_p}$, $\mathrm{C_{rp}}$ and $\mathrm{C_n}$ to achieve our objective. Intuitively, we should give a big value for $\mathrm{C_p}$ and a small value for $\mathrm{C_n}$ not only because their confidence is different, but also because the dataset in the problem of text classification is always unbalanced. The total number of positives is far less than that of negatives among the unlabeled example set. And the value of C_{rp} is between C_p and C_n because there are inevitable errors when extract positive examples from unlabeled examples.

For the vector d, the optimize problem (5) is 0-1 programming. On the other hand, it is a convex quad programming when d is given. These two programming can be resolved by turn. For convex quad programming, we can resolve its dual problem by introducing the lagrangian function. We have the following optimization problem (Due to space limitations, we do not list the detailed derivation).

$$\min_{d}\min_{\alpha,\beta}\ \frac{1}{2}\sum_{i=1}^{m+n}\sum_{j=1}^{m+n}(x_i\cdot x_j)(\alpha_i d_i-\beta_i(1-d_i))(\alpha_j d_j-\beta_j(1-d_j))$$

$$-\sum_{i=1}^{m+n}(\alpha_i d_i+\beta_i(1-d_i))$$

$$\text{s.t.}\ \sum_{i=1}^{m+n}(\alpha_i d_i-\beta_i(1-d_i))=0\,;$$

$$0\le\alpha_i d_i+\beta_i(1-d_i)\le c_p, i=1,\ldots,m\,;$$

$$0\le\alpha_i d_i+\beta_i(1-d_i)\le d_i c_{rp}+(1-d_i)c_n, i=m+1,\ldots,m+n \tag{6}$$

$$\sum_{i=1}^{n}d_i\le m+r\,;$$

$$d_i=1\,, i=1\,,\ldots,m$$

$$d_i=\{0,1\}\,, i=m+1\,,\ldots,m+n\,;$$

$$d=\{d_1,\ d_2,\cdots,\ d_{m+n}\}\,.$$

Where α_i, β_i $(i\ =\ 1,\ 2,\ldots,m+n)$ are Lagrange multipliers. After resolving the optimization problem (6) by iteration, we can obtain a more accurate SVM-based classifier.

3.2 Algorithm

Table 1 gives a detail process of resolving EB-SVM.

Table 1. EB-SVM algorithm discribed in Section 3.1

```
- Input: positive examples P, unlabeled examples U; $\delta = r/n$ ;
- Set $CP = \emptyset$, i=1;
- Assign $d_j = 0, j = m + 1, \dots, m + n$. Namely, each example in P the class label
  $+1$ and each example in U the class label $-1$;
- Loop;

    Use P and U to train a SVM classifier $C_i$ ;
    Classify U using $C_i$; Let the set in U that are classified as positive be $S$; the number
    of $S$ be $t$;
    If $r = 0$
        then exit-loop;
        else if $0 < t < r$
            then CP = S , $r = r - t$ ;
            else if $t > r$
              Sort decision values from $C_i$with descend. Then designate $r$ the top ranked
    examples $S_r$ as CP, $r = 0$;
              else exit-loop;
    $d_j = 1, \forall j$, saitisfy $x_j \in CP$; $P = P + CP$, $U = U - CP$, $i = i + 1$;

- Output: a text classifier $C_i$
```

4 Experiment

4.1 Experimental Setup

Datasets. 20Newsgroups[1] and Reuters[2] corpus are used to construct datasets. The 20newsgroups collection is the Usenet articles collected by Lang [5]. Each group has approximately 1000 articles. We use each newsgroup as the positive set and the rest of the 19 groups as the negative set, which creates 20 datasets. For Reuters corpus, the top ten popular categories are used. Each category is employed as the positive class, and the rest as the negative class. This gives us 10 datasets.

Preprocessing. In data pre-processing, we applied stop word removal, but no feature selection or stemming were done. Each document is represented as a vector of tf-idf value.

For each dataset, 30% of the documents are randomly selected as test documents. The remaining (70%) are used to create training sets as follows: δ percent of the

[1] http://www.cs.cmu.edu/afs/cs/project/theo-11/www/naive-bayes.html

[2] http://www.daviddlewis.com/resources/testcollections/reuters21578/

documents from the positive class is first selected as the positive set P. The rest of the positive documents and negative documents are used as unlabeled set U. We range δ from 10%-90% (0.1-0.9) to create a wide range of scenarios. For each training dataset, 30 percent of examples constitute the validation set.

In the experiment, the linear kernel function is used since it always performs excellently for text classification tasks [3]. We use LIBSVM[3] to build an SVM-based classifier for Biased-SVM and EB-SVM. LPU package[4] is used for the implementation of S-EM, ROC-SVM. Penalty factors are optimized on validation sets. The range of values for C_p, C_{rp} and C_n are from the set: $\{2^{-8}, 2^{-6}, \ldots, 2^{8}\}$ and final used values are auto-selected iteration index.

Performance Metric. We use the popular F score on the positive class as the evaluation measure. F score takes into account of both recall and precision

$$F = \frac{2pr}{p + r} \tag{7}$$

Where $r = TP/(TP + FN)$, $p = TP/(TP + FP)$. (TP and FP denote the number of true positive and false positive examples respectively. FN is the number of false negative examples).

F score cannot be computed on the validation set during the training process because there is no negative example. An approximate computing method [6] is used to evaluate the performance by

$$F = \frac{{r_p}^2}{\mathrm{Prob}(f(X) > 0)} \tag{8}$$

Where X is the random variable representing the input vector, $\mathrm{Prob}(f(X) > 0)$ is the probability of an input example X classified as positive, r_p is the recall for positive set P in the validation set.

4.2 Comparison with Biased-SVM

As shown in Fig. 1, EB-SVM outperforms Biased-SVM in most cases (δ from 0.1 to 0.8) on both corpora. The improvement is much larger for smaller δ because the number of positives in U is big and more samples from U will be identified as positives, potentially leading to a more accurate classifier for the next step yet. In other words, we can obtain very high precision on positive set P. Biased-SVM and EB-SVM obtain very similar performance when δ equals 0.9. This is because that the number of positives in U is very small and Biased-SVM obtains good performance by using all examples in U as negatives in this scenario.

[3] LIBSVM: `http://www.csie.ntu.edu.tw/~cjlin/libsvm`

[4] `http://www.cs.uic.edu/~liub/LPU/LPU-download.html`

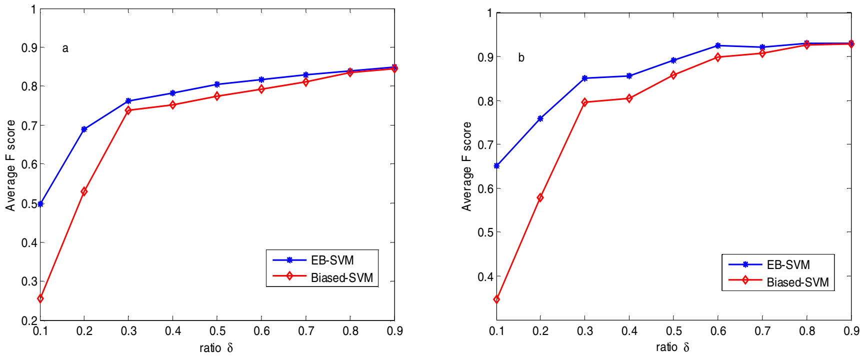

Fig. 1. Average F score comparision between EB-SVM and Biased-SVM on 20Newsgroups (a) and Reuters Corpus (b)

4.3 Comparison with Other Methods

Compared with other popular approach such as S-EM [10] and ROC-SVM [7] it can be seen from Fig. 2 that EB-SVM outperforms these methods in most δ on 20Newsgroups and Reuters corpora.

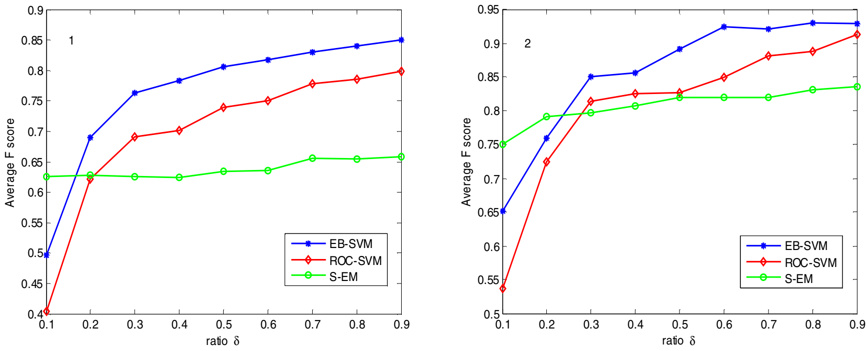

Fig. 2. Average F Score Comparison between EB-SVM and Other Methods on (1) 20Newsgroups and (2) Reuters Corpus

5 Conclusions

In this paper, we have put forward the extension algorithm based on Biased-SVM, called EB-SVM. Our developed approach is an iterative classification approach. The first iterative step is Biased-SVM and next to build a more accurate classifier by extracting confident positive from U. Experimental results have shown that EB-SVM can improve the performance of Biased-SVM.

References

1. Cortes, C., Vapnik, V.: Support vector network. J. Mach. Learn. 20, 273–297 (1995)
2. Fung, G.P.C., Yu, J.X., Lu, H., Yu, P.S.: Text Classification without Negative Examples Revisit. IEEE Transactions on Knowledge and Data Engineering 18(1), 6–20 (2006)
3. Joachims, T.: Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In: Nédellec, C., Rouveirol, C. (eds.) ECML 1998. LNCS, vol. 1398, pp. 137–142. Springer, Heidelberg (1998)
4. Manevitz, L., Yousef, M.: One-class SVMs for document classification. J. Mach. Learn. Res. 2, 139–154 (2001)
5. Lang, K.: Newsweeder: Learning to filter netnews. In: Proceedings of the 12th International Machine Learning Conference, Lake Tahoe, US, pp. 331–339 (1995)
6. Lee, W.S., Liu, B.: Learning with Positive and Unlabeled Examples Using Weighted Logistic Regression. In: Proceedings of the 20th International Conference on Machine Learning, Washington, DC, United States, pp. 448–455 (2003)
7. Li, X., Liu, B.: Learning to Classify Text Using Positive and Unlabeled Data. In: Proceedings of the 18th International Joint Conference on Artificial Intelligence, Acapulco, Mexico, pp. 587–594 (2003)
8. Li, X.-L., Liu, B., Ng, S.-K.: Learning to Classify Documents with Only a Small Positive Training Set. In: Kok, J.N., Koronacki, J., Lopez de Mantaras, R., Matwin, S., Mladenič, D., Skowron, A. (eds.) ECML 2007. LNCS (LNAI), vol. 4701, pp. 201–213. Springer, Heidelberg (2007)
9. Li, X., Liu, B., Ng, S.: Negative Training Data can be Harmful to Text Classification. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, Massachusetts, USA, pp. 218–228 (2010)
10. Liu, B., Lee, W.S., Yu, P.S., Li, X.: Partially Supervised Classification of Text Documents. In: Proceedings of the 19th International Conference on Machine Learning, Sydney, Australia, pp. 387–394 (2002)
11. Liu, B., Dai, Y., Li, X., Lee, W.S., Yu, P.S.: Building Text Classifiers Using Positive and Unlabeled Examples. In: Proceedings of the 3rd IEEE International Conference on Data Mining, Melbourne, Florida, United States, pp. 179–188 (2003)
12. Nigam, K., McCallum, A.K., Thrun, S.: Learning to Classify Text from Labeled and Unlabeled Documents. In: Proceedings of the 15th National Conference on Artificial Intelligence, pp. 792–799. AAAI Press, United States (1998)
13. Nigam, K., McCallum, A.K., Thrun, S., Mitchell, T.: Text Classification from Labeled and Unlabeled Documents Using EM. Mach. Learn. 39, 103–134 (2000)
14. Sebastiani, F.: Machine Learning in Automated Text Categorization. ACM Computer Surveys 34, 1–47 (2002)
15. Yu, H., Han, J., Chang, K.C.C.: PEBL: Positive Example-Based learning for web page classification using SVM. In: Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 239–248. ACM, United States (2002)